

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Приладобудівний факультет**

**Кафедра приладів і систем орієнтації і навігації**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Бурау Н. І.

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

## **Дипломний проект**

**на здобуття ступеня бакалавра**

**за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології**

**на тему: «Вдосконалення сайту кафедри ПСОН»**

Виконав:

студент III курсу, групи ПГ-пб1

Подройко Сергій Вікторович \_\_\_\_\_

Керівник:

Доцент к. т. н. Павловський О. М. \_\_\_\_\_

Консультант з: \_\_\_\_\_

Рецензент:

Доцент к. т. н. Маркіна О. М. \_\_\_\_\_

Засвідчую, що у цьому дипломному  
проекті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_

Київ – 2019 року

## АНОТАЦІЯ

Об'єм пояснювальної записки 67 сторінок. У роботі представлено 19 графічних рисунків та 2 таблиць. Кількість літературних посилань - 23. Мета роботи – удосконалення сайту кафедри приладів та системи орієнтації та навігації. Для реалізації поставленою мети поставлені наступні задачі:

- аналіз фреймворків та засобів розробки сайту кафедри ПСОН;
- створення структури сайту;
- створення дизайну сайту;
- SEO-оптимізація сату
- Удосконалення Web-сайту та наповнення його матеріалами.

Ключові слова: Сайт, HTML, CSS, CMS, Google, bootstrap, SEO.

## ANNOTATION

The volume of the explanatory note is 67 pages. The paper presents 23 graphic drawings and 2 tables. The number of literary sources is 17. The purpose of the work is to improve the site of the chair of devices and the system of orientation and navigation. To achieve this goal set the following tasks:

- analysis of bases and means of development of the site of the department of PSD;
- creating a site structure;
- creation of site design;
- SEO-optimization of the site;
- Improve the website and fill it with materials.

Keywords: Website,HTML, CSS, CMS, Google, bootstrap, SEO.

## Зміст

Вступ.....	9
1 АНАЛІЗ САЙТІВ ОСВІТНЬОГО НАПРЯМКУ.....	11
1.1 Поняття веб-сайту.....	11
1.2 Історія заснування сайту.....	12
1.3 Обґрунтування потреби в веб-сайті.....	13
1.4 Аналіз існуючих сайтів кафедр «КПІ імені ІГОРЯ СІКОРСЬКОГО».....	14
1.5 Структурна веб сайту.....	18
1.5.1 Огляд Front – End фреймворків.....	21
1.5.2 Переваги і недоліки застосування Front-end фреймворків.....	25
1.5.3 Back - End фреймворків та критерії вибору фреймворку для Back – End.....	26
1.5.4 Огляд Back-end фреймворків.....	26
1.5.5 Загальні переваги використання Back-end фреймворку.....	34
1.5.6 Недоліки застосування Back-end фреймворку.....	35
1.5.7 Web-сервер Apache.....	36
1.5.8 Шляхи та області використання.....	37
1.6 Обґрунтування вибраного веб-редактора.....	38
1.7 Висновок до розділу 1.....	38
2. ВИБІР ФРЕЙМВОРКУ ДЛЯ РЕАЛІЗАЦІЇ. ОГЛЯД РОБОТИ З ОБРАНИМ ФРЕЙМВОРКОМ. SEO-ОПТИМІЗАЦІЯ САЙТУ.....	39
2.1 Вибір Front-end фреймворку для реалізації сайту2.1.1 Bootstrap структура .....	39
2.1.1 Bootstrap структура.....	39

2.1.2 Класи для комірок.....	41
2.2.1 Процес розробки сайту з використанням Bootstrap.....	41
2.2. SEO-ОПТИМІЗАЦІЯ САЙТУ.....	42
2.2.1 Види SEO-оптимізації.....	42
2.2.2 Методи SEO.....	43
2.2.3 Мета-теги.....	44
2.2.4 Вимоги до SEO оптимізації.....	45
2.2.5 Додатки для оптимізації.....	47
2.3 Висновки розділу 2.....	50
3. УДОСКОНАЛЕННЯ САЙТУ ЗА ДОПОМОГОЮ BOOTSTRAP ТА SEO ОПТИМІЗАЦІЯ. СТВОРЕННЯ НОВОГУ САЙТУ НА CMS ПЛАТФОРМІ.....	51
3.1 Загальна структура сайту.....	51
3.2 Удосконалення інтерфейса.....	52
3.3 Удосконалення SEO-оптимізації сайту.....	54
3.4 Створення сайту на CMS.....	56
3.4.1 Популярні CMS платформи.....	58
3.5 Створення сайту за допомогою WordPress.....	59
ВИСНОВКИ.....	65
ПЕРЕЛІК ПОСИЛАНЬ.....	66

## ВСТУП

В наш час, більшість людей має потребу у пошуку потрібної інформації. Насамперед, це стає найбільш актуальним в роки інформаційного перевантаження, тобто у сьогоденні. Перевантаження інформацією – це термін, який має на увазі ті самі труднощі для розуміння проблем і шляхів їх вирішення - все це є наслідком неабиякого надлишка інформації. Потрібно звернути увагу, що концепція цього терміну була розроблена у далеких 70-х роках 20-го століття. Ця проблема почала стосуватися більшість людей, коли стрімко зріс інтерес суспільства до Інтернету, тоді це стало по-справжньому актуальним. Як ми добре знаємо, саме всесвітня павутина є джерелом невичерпної інформації, яка є доступною для кожного користувача.

Процеси, в яких фігурує суспільний розвиток, призводять до зрозумілих наслідків - зміни індустріального суспільства на інформаційне. Створені технології стають визначальними у житті, починають впливати у на всі аспекти життя, на суспільний розвиток, на будь-яку діяльність та її процеси. Насамперед, розвиток сфери інформаційних технологій у нас час є тим неодмінним фактором, який впливає на фундаментальні рішення в головних ланках - економіці та політиці, культурі та освіті. Варто розуміти те, що саме через ці чинники, такий ресурс як інформація набуває стратегічного значення, в той же час, коли загалом інформаційні технології ми бачимо, як його інструмент. Потрапляючи у безмежний простір, який має освіта, створені технології зовсім різного характеру є причиною до корінних змін у користь популярних тенденцій у суспільстві. Насамперед, це робить процес знаходження інформації набагато ефективнішим та швидким, бо впливові місця займають саме ресурси та бази, які надають нам змогу фільтрувати данні, прискорюють вирішення можливих проблем, що, у свою чергу, є головним у вічній галузі - освіті.

Кожна організація, підрозділ вищого закладу забезпечує у своїй діяльності декілька головних функцій. Наприклад, представницька, що

забезпечує зберігання наукових робіт, історичних та цінних відомостей про життя. Інформаційна функція стосується новин діяльності. Звісно, освітня об'єднує дані у легкому доступі, що полегшує навчання: матеріали, відомості, розклади тощо. Також варто згадати ще одне - комунікації, тобто надання студентам можливість ставити питання до адміністрації вищого навчального закладу, викладачів, інших користувачів. З усього цього стає зрозумілим, що є для установ пріоритетним. Конкурентоспроможність, гарний імідж, ефективність роботи зі своєю інформацією та доступність - усе це є головним для закладу.

По-перше, сторінка кафедри є ресурсом не тільки для студентів, а ще для викладачів, працівників, абітурієнтів, діячів, які цікавляться історією та діяльністю галузі, тих, хто має зацікавленість у навчальному процесі, наукових робіт та здобутках розглянутого університету.

Отже, робота, яка полягає у створенні надійного сайту кафедри, є актуальною, бо забезпечує виконання на високого рівні всіх функцій, які можливо надати інформаційно: формує імідж, тримає багато корисної інформації, дає змогу спілкуватися відвідувачам та ставити запитання.

Мета для підвищення рівня сайту - просування сайту у Інтернеті, розробка та створення структури, збільшення ефективності для користувачів.

З цього можливо сказати, що для досягнення мети, ми потребуємо виконати наступні завдання:

- аналізувати доступні методи та засоби створення сайту;
- забезпечити адміністрацію можливістю авторизуватися у системі;
- створити належну структуру роботи сайту кафедри;
- зайнятися просуванням ресурсу у пошукових системах;
- розробити Web-сайт та опрацювати інформацію для подальшого заповнення матеріалами.

# 1 АНАЛІЗ САЙТІВ ОСВІТНЬОГО НАПРЯМКУ

## 1.1 Поняття веб-сайту

Веб-сайт (web — «павутина», «мережа» і site — «місце», website - «місце в мережі») або просто сайт — це сукупність електронних документів (файлів) приватної особи або організації, що об'єднана під однією адресою (доменною або IP-адресою).[1]

Насамперед, варто зауважити, що, за винятком веб-сайтів, існують також WAP-сайти, тобто тих, що направлені для використання на мобільних телефонах та комп'ютерах. Раньше, коли головною діяльністю Мережі та Інтернета взагалі були розваги, це трактування було правильним. Цікаво те, що ще в 20 столітті, до кінці 90-х, макети веб-сайтів були зовсім статичні. Головним знанням для створення нової сторінки був HTML - мова гіпертекстової розмітки, яка не завдає труднощів для оволодіння на даний момент. Будь-який функціонал, який мав даний ресурс, був лише додатковим плюсом, використаного для розміщення, сервера. Якщо згадати про естетику та зручність тогочасного дизайну, то її просто не було.

З плином часу, як ми знаємо, знання розповсюджуються по всьому світу. Сьогодні більшість організацій та установ, підприємств та закладів, фірм стали повноважним населенням Інтернету - створили власні сторінки.

Для різних типів послуг, що надають веб-портали (великі сайти), існують різні домени, які використовуються в залежності від наданих там послуг. Наприклад, mail.google.com, news.google.com, maps.google.com тощо. Також любляють створювати домени для різної територіальної громади (google.com.ua і google.ru - це сайти великої корпорації Google, яка надає користувачам можливість відвідувати ресурси власної мови, але технічний аспект сторінок різний).



Безкоштовні хостинги, наприклад, характеризуються об'єднанням декількох сайтів під єдиним доменним ім'ям. У першу чергу для цього використовують домени третього рівня, приміром слугує `mysite.example.com`. В той же час для уособлення сайтів, їх ідентифікації, вказується і надане ім'я сайту, але після доменної адреси хосту та тильди (`example.com/~my-site-name/`).

Веб-сервери - це потужні ресурси, які створюються для зберігання великої кількості даних: файлів, зображень, аудіофайлів, відеофайлів и т.п.. Ця послуга у збереженні інформації називається веб-хостингом. Зберігання всього на індивідуальному сервері було популярним раніше, але розвиток інформаційних технологій спричинив можливості для розміщення декількох сайтів разом. Однак, також не перестають існувати сервери тільки для одного сайту.

## **1.2 Історія заснування сайту**

Самий перший сайт `info.cern.ch` з'явився в 1990 році. Його автор, Тім Бернерс-Лі, опублікував на ньому опис нової технології World Wide Web, яка базується на протоколі передачі даних HTTP, системі адресації URL і мові гіпертекстової розмітки HTML. Також, на сайті було описано принципи установки і роботи серверів та браузерів. Сайт став і першим в світі Інтернет-каталогом, бо згодом Тім Бернерс-Лі розмістив на ньому список посилань на інші сайти.[2]

Бернерс-Лі підготувався та створив все необхідне для вдалої роботи свого першого сайту ще з самого початку. Наприклад, у 1990 році світу був представлений перший гіпертекстовий браузер WorldWideWeb, який мав функції для веб-редактора. Також ця особа піклувалась про сервер, створивши його, використовуючи NeXTcube, а ще - веб-сторінки, які були єдиними на той час. Тобто Тіма Бернерс-Лі можна окреслити справжнім «батьком» базових технологій Веб — HTTP, URI/URL і HTML.

Цей винахідник зумів втілити свою ідею у життя та заснувати гіпертекст, який став головною структурною ланкою для мереж по обміну інформації. Enquire - та сама гіпертекстова розробка, що була їм створена Тімом у далекому 1980 році. Воно було побудовано на технології зберігання даних шляхом випадкових асоціацій. Через роки праці в CERN (Європейський центр ядерних досліджень, що розташований в Женеві) Тім Бернерс-Лі заохочує інших працівників представити документи на гіпертексті, що об'єднані багатьма гіперпосиланнями. З того часу було знайдено можливість забезпечити доступ до внутрішніх даних документів та первинних джерел Всесвітнього павутиння. Травень 1991 року став визначним саме тому, що в цього місяця було ухвалено вічний стандарт WWW у Женеві

На даний час Тім Бернерс-Лі очолює заснований ним Консорціум Всесвітньої павутини (World Wide Web Consortium), який займається розробкою і впровадженням стандартів Інтернету.[3]

### **1.3 Обґрунтування потреби в веб-сайті**

Не є секретом те, що прогрес галузі інформаційних технологій використовують будь-де. Він став невід'ємною складовою для навчання та створення функціоналу для швидкого користування наданою інформацією та забезпечення соціально зв'язку. У наш час, напевно, неможливо знайти підприємство, організацію чи заклад, що не має посилань на свій сайт, бо він став доступною можливістю надавати інформацію іншим. Створити сайт це рішення для поставлених задач:

- Донесення нової інформації;
- Інформування користувачів;
- Створення розділів сайту;
- Надання контактів;
- Інформування про педагогічний склад кафедри;

- Опублікування даних для студентів закладу.

Тобто створений ресурс потребує у можливості надання достовірних матеріалів для майбутніх студентів, розповідати про викладачів, надавати необхідну інформацію.

#### 1.4 Аналіз існуючих сайтів кафедр «КПІ імені ІГОРЯ СІКОРСЬКОГО»

Розглядаючи «КПІ імені ІГОРЯ СІКОРСЬКОГО» ми знайшли 132 сайтів. велика кількість з цього переліку було зроблено на CMS (Система керування вмістом (СКВ; англ. Content Management System, CMS) — це необхідне програмне забезпечення, головною функцією якого є створення зручного функціоналу для будь-яких ресурсів або мережах. Сторінки, які застосовували технології CMS, прості для використання та оновлення інформації. Зауважимо, що у «Рейтингу сайтів КПІ ім. Ігоря Сікорського» кафедра ПСОН посіла 74 місце (Рисунок 1.1) за січень 2019 року.

Місце	Місце на 10.18	Зміни	Кафедра	Домен	факультет	Google-Index	Місце - Index (40%)	Ahref	Majestic	Місце VISIBILITY (50%)	Google-Scholar Citation	Місце Citation (10%)
74	51	-23	Кафедра приладів та систем орієнтації і навігації	kafpson.kpi.ua	pbf	549	91	17	11	32	339	89

Рисунок 1.1 – рейтингова таблиця сайтів КПІ ім. Ігоря Сікорського

Для сайтів інших учасників маємо: ФБТ, «Кафедра технічної кібернетики» факультет ФІОТ, факультет ПСНК -вони дуже схожі, бо мають інформаційний характер: дані про себе, різні новини та заходи. Ці сайти можна виділити як «статичні», бо перевага надається текстовому формату.

До користі «статичних» сайтів для роботи кафедр відносять:

- наявність подібних сторінок - це можливість використовувати один шаблон, але з різним стильовим оформленням ; Змін...Лист № докум. Підпис Дата Лист 14 ІМ11.15.1300.1118.ПЗ
- швидке кешування сторінок;

- можливість простого переноса на інший сервер;
- простота в оформленні;
- легкість редагування інформації. Недоліки «статичних» сайтів науково-практичних конференцій:
  - Немає можливість займатись важким контентом, тобто щоб відреагувати інформацію треба прописувати HTML-код;
  - Щоб наповнити новою інформацією необхідно треба отримати FTP доступ сайту;
  - Не має можливості реєстрації. Сайти «статичних» кафедр несуть всю необхідну інформацію про новини та події кафедр. Вони не мають управлінням над файлами, реєстрацією, та надання прав зареєстрованим юзерам. Сайти кафедр «технічної кібернетики» та «приладів і систем неруйнівного контролю» мають багатомовний інтерфейс, що є плюсом, тому що інострани студенти зможуть ознайомитись з інформацією про факультет.

Для підвищення місця в рейтингу використовували Google Analytics.

Google Analytics (GA) – використовується для деталізації даних статистики про відвідувачів веб-сайтів, це безкоштовний сервіс, створений Google, який показує детальну статистику. Також сервіс надає можливість оцінити кількість дій: конверсій та продажів, новий погляд на використання сайту відвідувачами. Він буде корисний для пошуку джерела трафіку та порад (Рисунок 1.2).

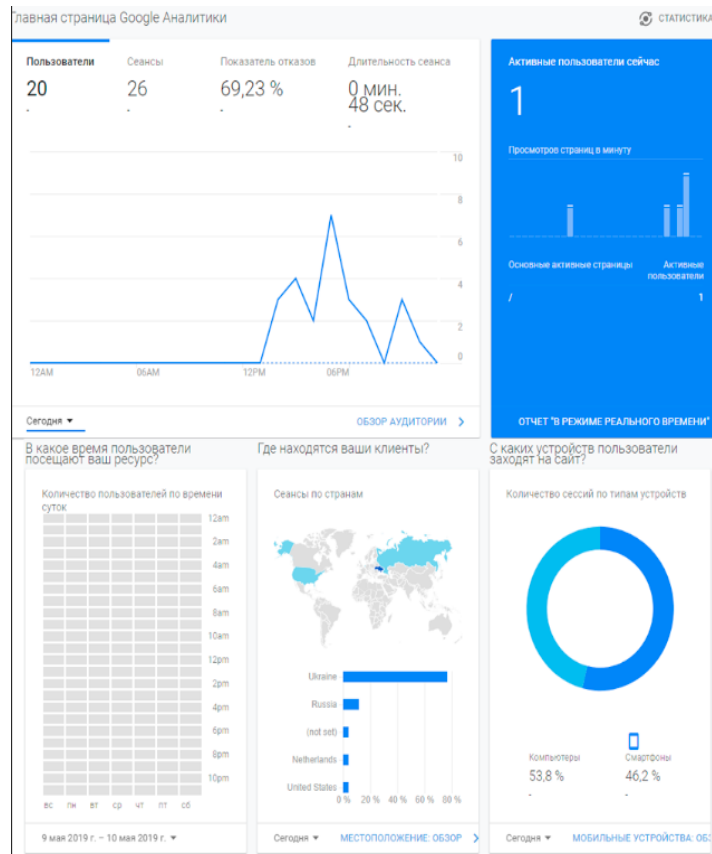


Рисунок 1.2 – Інформаційна панель платформи GA

Більшість свого часу треба використати на аналіз інформації про швидкість сайту, середню вартість замовлення та коефіцієнт конверсії (характерно для тих підприємств, які продають товари, послуги).

Щоб відразу побачити, що працює ефективно, а що – ні, можна відстежувати активність на своєму сайті (Рисунок 1.3). Таким чином ви зможете кожного дня бути в курсі того, що відбувається з компанією.

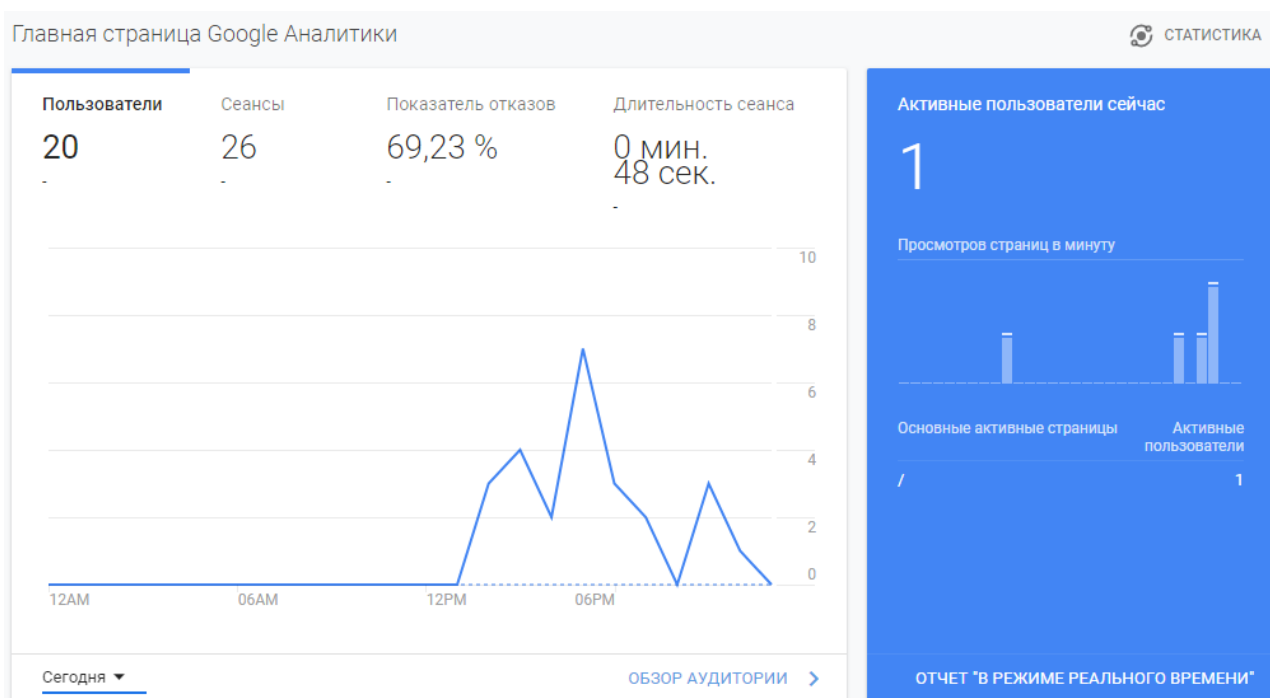


Рисунок 1.3 – Панель огляду активності на платформі GA

Список функцій GA:

1) Наявні інструменти для роботи. GA побудовано на основі потужної, але простої у користуванні платформи для звітів. Це створено для того, щоб ви могли вирішувати, що саме переглядати, а також змінювати, налаштовувати звіти за декілька кліків.

2) Аналіз вмісту. Наявні інформаційні дані - звіти, вони виявляють дійові частини сайту, показують найпопулярніші посилання.

3) Соціальний аналіз. Так як мережа наповнена соціумом, то Google Analytics вміє оцінювати корисну дію ваших програм, які є соціальними засобами комунікації. На цьому етапі можна проаналізувати взаємодії відвідувачів та функції, які вони використовують для обміну інформації на сайті (наприклад, кнопка +1 у Google), вашими даними у соціальних мережах (Рисунок 1.4).

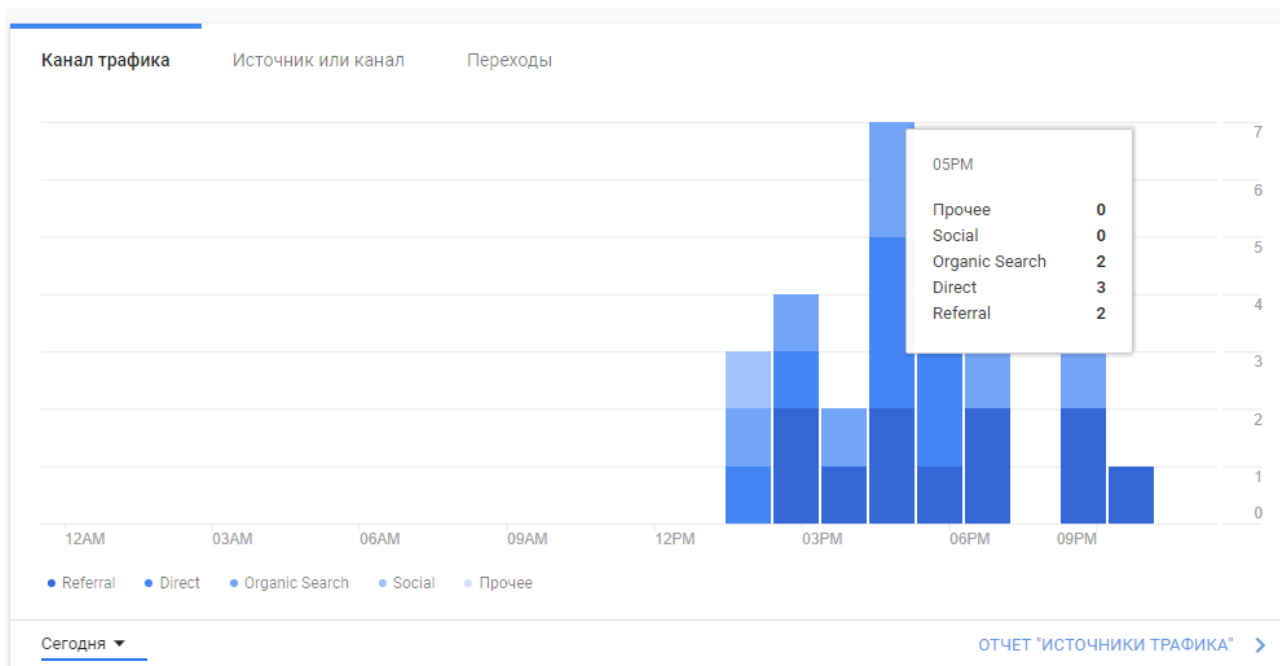


Рисунок 1.4 – Панель соціальної активності на платформі GA 16

4) Мобільні пристрої. Також Google Analytics оцінює, насамперед, вплив мобільної мережі на ваш проект. Поза цим, він створює програми для мобільних пристроїв, Google Analytics пропонує вам розробки програмного забезпечення для платформ на iOS і Android, тому ви можете оцінити, як саме використовується ваша програма.

## 1.5 Структурна веб сайту

Головним етапом розробки сайту для веб-майстрів є верстка. Раньше ми вже зустріли докладний опис цього процесу. Тепер ми розглянемо саму верстку, її особливості та типи. Без неї не може функціонувати жоден сайт, тому прийнято виділяти це в спеціалізацію веб-майстрів - верстальник. Під назвою “верстка” йдеться про процес перекладу якогось графічного макету, який було створено раніше, в цілі веб-сторінки, щоб далі розмістити їх у мережі інтернет. Саме завдяки виробленій специфіці верстки, можна регулювати так звану еластичність сайту, розбити адаптивність для ресурса - коректного відображення у

пошукових системах різного типу і на різних пристроях. Таким чином верстка робить початок, завдає специфіку майбутнього етапу створення.

треба не тільки зробити гарний шаблон, бо процес створення сайту набагато складніше та довше. Важливо навчитися правильно використовувати готовий дизайн при розробці майбутнього сайту. Справа в тому, що новий зверстаний шаблон інтегрується зі сторінками сайту за допомогою мов програмування. І тут починається, напевно, один зі складніших етапів роботи. Адже від того, наскільки вміло веб-майстер володіє мовою програмування залежить якість, комфортність та надійність в роботі зверстаного їм сайту.

Веб-майстри, насамперед, бачать наступні три типи макета верстки:

- Розширювана верстка. Після версток сайти будуть відображатися задовільно у всіх типах браузерів, але є ризик виникнення можливих проблем в процесі роботи.
- Шаблонна верстка - суворе дотримання розміру рядка, наданих параметрів.
- «Гумова» або «пристосована» верстка забезпечує відмінне відображення на всіх моніторах, незалежно від якості дозволу.

До покращення сайту виглядів так як на рисунку, (рисунок 1.5), побудований був на табличній верстки, розмітка виглядала таким так як на (рисунку 1.6), також не було гумовості сайту.



```

1 <html>
2 <head>
3 <title>Кафедра ПСОН, Приборостроительный факультет, НТУУ "КПИ"
4 </title>
5 <meta http-equiv="content-type" content="text/html; charset=windows-1251">
6 <meta name="keywords" content="Кафедра ПСОН, навигационная система, гироскоп, приборы ориентации, цифровые системы, Одинцов А.А., Бурау Н.И., гироскоп">
7 <link rel="stylesheet" href="css.css" type="text/css">
8 </head>
9 <body bgcolor="#FFFFFF" text="#000000" leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
10 <table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
11 <tr>
12 <td width="360" height="132" valign="bottom"></td>
13 <td width="100%" valign="top" align="right" background="diz/line-big.jpg">
14 <table width="100%" border="0" cellspacing="0" cellpadding="0" height="132" align="right">
15 <tr>
16 <td background="diz/line-big.jpg" height="132" valign="top" align="center" width="100%" class="hi">
17 <script language=JavaScript>
18 <!--
19 if (navigator.appName=="Microsoft Internet Explorer"){
20 document.writeln
21 ('<div style="position:absolute; left:271px; top:5px; width:430px; height:50px; z-index:1; overflow: visible; visibility: visible"> <font size="5">
22 } else {
23 document.writeln(' ')
24 } //-->
25 </script>
26 </td>
27 <td height="132" width="17" valign="top"></td>
28 </tr>
29 </table>
30 </td>
31 <td colspan="2">
32 <table width="100%" border="0" cellspacing="0" cellpadding="0" height="132">
33 <tr>
34 <td height="92" valign="top"></td>
35 </tr>
36 <tr>
37 <td height="40" width="100%" valign="top">
38 <table width="100%" border="0" cellspacing="0" cellpadding="0" height="40" align="left">
39 <tr>
40 <td background="diz/30a.jpg" height="40" width="100%">

```

Рисунок 1.6 – html код табличної верстки

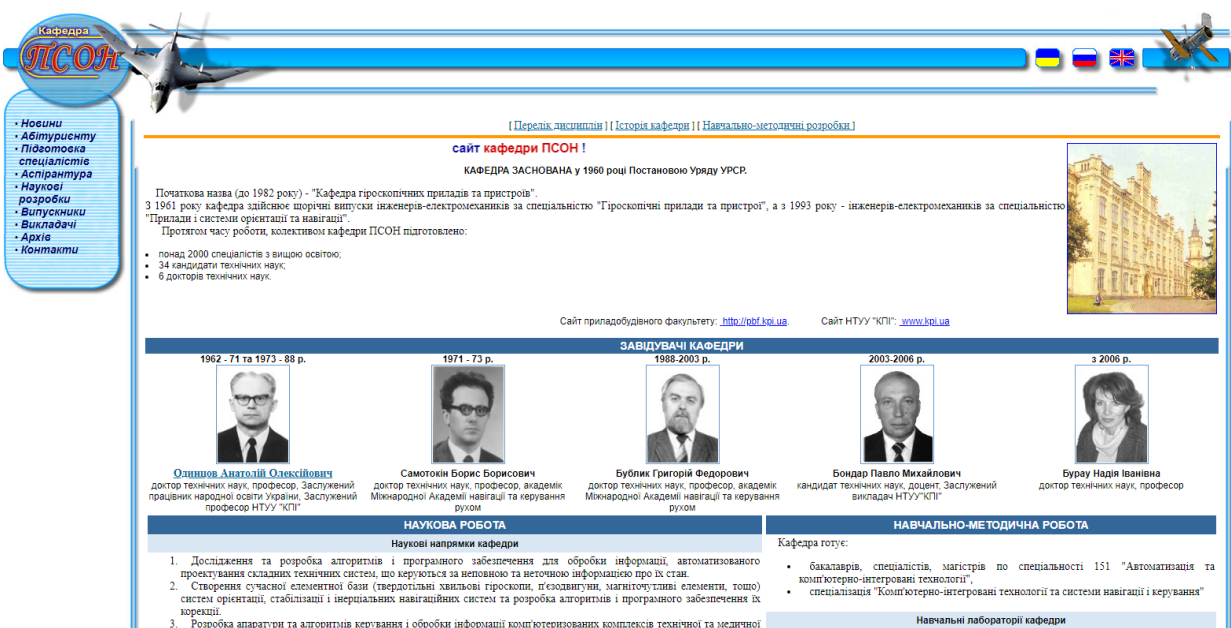


Рисунок 1.5 – головна сторінка сайту до удосконалення

Для удосконалення сайту використали такий фреймворк як Bootstrap.

Фреймворк - інфраструктура для програмних рішень, яка полегшує розробку навіть найскладніших систем. Спрощена дану інфраструктуру можна вважати своєрідною комплексною бібліотекою, але при цьому вона має ряд обмежень, які задають правила створення структури проекту та написання коду.

### 1.5.1. Огляд Front – End фреймворків

**Twitter Bootstrap** - це безкоштовний набір інструментів з відкритим вихідним кодом, призначений для створення веб-сайтів та веб-застосунків, він містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Bootstrap спрощує розробку динамічних веб-сайтів і веб-застосунків.[4]

Він, по-перше, є дуже відомим CSS-фреймворком. На ньому створено багато мільйонів проектів і в майбутньому буде створено точно не менше. Напевно, ця надзвичайна популярність криється в дуже ранньому старті, або не останню роль має доволі шляхетне походження (проект створювався в компанії Twitter). У кожному випадку Bootstrap вважається одним із найпопулярніших фреймворків та не здає позицій.

Я вважаю, що головною в цьому фреймворку є сітка, бо останнє оновлення Bootstrap підтримує адаптивність навіть у 12-ти колонкову сітку. Мобільні пристрої також не забуті. З третьої версії фреймворку починає діяти підхід “Спочатку мобільні”(розробляється версія сайту для мобільних пристроїв).

UI набір елементів в Bootstrap є одним із найбільших. Наявні елементи керування на всі будь-які випадки життя. Якщо чогось немає, то в більшості випадків необхідне доповнення можна легко знайти в мережі. Ця популярність робить свою справу, тому моменти, коли ви чогось не зможете знайти для bootstrap зводяться до мінімуму.

Bootstrap рівномірно відноситься до любителів Sass і Less. Вже після виходу релізу CSS на LESS з'являється порт на Sass.

Розробники Bootstrap, напевно, розуміють, що цей проект виріс і навряд чи всі його компоненти, можливості будуть популярні в межах одного додатка. Для того, щоб не тягнути за собою непотрібні файли, на офіційному сайті є можливість зібрати свій дистрибутив Bootstrap, включивши в нього тільки необхідні елементи. До переваг та недоліків цього фреймворку наплетимось:

#### Переваги:

- гарна документація;
- адаптивна побудова сайту;
- кроссбраузерність;
- швидке створення сайту.

#### Недоліки:

- велика популярність створює багато сайтів однакового вигляду, дуже складно створити унікальний дизайн;
- велика кількість класів;
- розмір фреймворку великий, це дуже сповільнює завантаження сторінки.

**Zurb Foundation** – це один з найпопулярніших фреймворків, для створення адаптивного дизайну. За своєю популярністю цей фреймворк не поступається Bootstrap. Кількість зірок на GitHub зайвий раз підтверджує довіру розробників до цього фреймворку. Foundation був створений в компанії Zurb (звідси і назва), що спеціалізується на створенні дизайнів веб-додатків. Створювався проект, в першу чергу, для власних потреб, тому з коробки націлений вирішувати реальні завдання. [5]

Foundation пропонує нам велику колекцію готових компонентів (віджетів), 12-ти колонкову сітку та ще декілька звичних речей. Бібліотека UI компонент має солідний вигляд. Тільки її вистачить для роботи над реальними прототипами. Співтовариство Foundation створилося дуже давно, тому проблем з пошуком додаткових плагінів/інформації просто немає. З четвертої версії, Foundation

виводить на передній план принцип «спочатку мобільні». У випадку з Foundation ц не говорить про повну адаптивність і відмову від застосування бібліотеки jQuery на користь zepto.js. З його особливостей варто відзначити схожість з Bootstrap, саме через це при наявності досвіду перейти на новий фреймворк буде в рази легше. Тобто, переваги та недоліки цього фреймворку:

Переваги:

- Наявна валідація форм;
- Підтримка RTL;
- Легкість у розширенні функціоналу, бо існує велика кількість доповнень;
- Хороша підтримка мобільними пристроями.

Недоліки:

- Високий поріг входження.

**Semantic UI** - фреймворк для побудови призначеного для користувача інтерфейсу веб-додатку з широкими можливостями по налаштуванню і “повне право вільно вирішувати дизайн”, як обіцяють самі розробники. Зараз набір включає 50+ елементів UI і 3000+ змінних CSS, але колекція швидко поповнюється. Фреймворк повністю адаптивний, всі компоненти визначаються за допомогою ems або rems. [6]

Один з головних наголосів у фреймворка це семантичність класів. Компоненти, які пропонує Semantic UI включає кнопки, зображення, мітки, спливаючі вікна, анімовані кнопки, роздільники, значки, сегменти, кастомізовані checkbox'и і radio, text loader'и тощо. Ще неможливо не помітити наявність доволі цікавих прийомів. Наприклад, можливість використовувати мітки, для того , щоб відобразити «стрічку» в кутку елемента.

До додаткових особливостей фреймворка належать можливість додати 'Disabled' стан картинкам і піктограмам, в них можуть бути відформатовані різні

кольори і розміри. Ряд популярних іконок включений за допомогою сервісу Fontello.

Можливості:

- Проста у використанні;
- Різноманітність компонентів;
- Теми;
- Семантичний;
- Має дуже активний розвиток;
- Має більше можливостей, ніж бутстрап;
- компоненти для користувацького інтерфейсу.

Недоліки:

- Не підтримує препроцесори SASS, LESS;

Кожен з усіх розглянутих фреймворків має свої переваги та недоліки. Тому вибір фреймворка потрібно робити з урахуванням проекту, над яким ведеться робота. Це - головний критерій вибору, бо жоден з фреймворків не є універсальним рішенням для будь-якого проекту. Тут ви можете побачити таблицю порівняння основного функціоналу вищерозглянутих фреймворків.

Таблиця 1.1 – Порівняльна характеристика популярних Front-end фреймворків для створення адаптивного дизайну.

Можливості	Bootstrap	Zurb Foundation	Semantic UI
Препроцесори	LESS,SASS	SCSS	-
JS-бібліотеки	Jquery	Jquery	Jquery
Підтримка плагінів	-	Modernizr, FastClick	-
Адаптивна сітка	+	+	+
Слайдери	+	+	-
Випадаючі списки	+	+	+
Навбар	+	+	+
Canvas меню	-	+	-

Таблиця 1.1 - Порівняльні характеристик

### 1.5.2. Переваги і недоліки застосування Front-end фреймворків

Взявши розглянутий вище матеріал, можна виділити такі переваги й недоліки використання CSS-фреймворків у своєму проєкті:

Переваги:

- Блочна верстка;
- Ретельно підібрані стилі покращують читабельність контенту;
- Кроссбраузерність;
- Пришвидшення розробки;
- Одноманітність коду дозволяє знизити кількість конфліктів при командній розробці;
- Допомогає недосвідченому розробнику правильно створювати HTML-макет;
- Дає можливість використовувати генератори коду.

Недоліки:

- Залежність від сторонньої бібліотеки; 37
- Деякі фреймворки використовують префікси, що ускладнює розуміння коду;
- Великий розмір фреймворків. Як можна побачити CSS, фреймворки мають невелику кількість недоліків у порівнянні з їх перевагами.

### **1.5.3. Back - End фреймворків та критерії вибору фреймворку для Back – End**

Для Back-end також існують сотні чи тисячі фреймворків, які є реалізованими на зовсім різних мовах програмування, таких як: Java, Rubi, PHP, Python, на технології створення веб-додатків, веб-сервісів від компанії Майкрософт ASP.NET. Далі буде розглянуто найпопулярніші з них.

#### **1.5.4. Огляд Back-end фреймворків**

**Yii** – це універсальний фреймворк, який може бути задіяний у всіх типах веб додатків. Завдяки його компонентній структурі та відмінній підтримці кешування, фреймворк особливо підходить для розробки таких великих проектів як портали, форуми, CMS, магазини чи RESTful-додатки.[7]

Можливості Yii :

- Низький поріг входження
- Парадигма Модель-вид-контролер
- Інтерфейси DAO і ActiveRecord для роботи з базами даних (PDO)
- Можливість інтернаціоналізації
- Кешування сторінок і окремих фрагментів
- Перехоплення і обробка помилок
- Введення і валідація форм
- Аутентифікація і авторизація
- Використання AJAX і інтеграція з jQuery
- Генерація базового PHP-коду для CRUD-операцій (скаффолдинг)

- Підтримка тем оформлення для їх легкої зміни
- Можливість підключення сторонніх бібліотек
- Міграції бази даних
- Автоматичне тестування
- Підтримка REST

Недоліки:

- AR не підтримує AR запити
- Не дуже гарний зв'язок в БД багато-до-багатьох(плагін CAdvancedArBehavior виправляє цей недолік)

**Symfony2** – PHP фреймворк, який має велику бібліотеку класів, написаний на PHP 5. Архітектура має корисні компоненти та інструменти, призначені для створення складних веб-додатків. Symfony — це вільний каркас, написаний на PHP5, який використовує патерн Model-View-Controller (MVC). Symfony пропонує швидку розробку і керування веб-додатками, що дозволяє легко вирішувати рутинні завдання веб-програміста. Symfony безкоштовний і доступний під ліцензією MIT.[8]

Перваги :

- Підтримує безліч баз даних (MySQL, PostgreSQL, SQLite, або будь-яка інша PDO-сумісна СУБД)
- Вбудовані класи для роботи з email
- Дуже гнучка система шаблонів у поданні
- Вбудований код-генератор
- Підтримка французького спонсора Sensio
- Гнучкий
- Має високу продуктивність

Недоліки:

- Складний для освоєння
- Підходить тільки для великих проектів



- Немає російської документації
- Остання версія 5.3 вимагає PHP
- Не забезпечений вбудованою ORM
- Відсутність російськомовного співтовариства

**Zend framework** – це PHP– фреймворк, що створений і підтримується компанією Zend, співробітники якої є безпосередніми авторами мови PHP. Тому він є послідовником традицій і цінностей PHP – базується на простоті, об'єктноорієнтованих принципах, дружній ліцензії і ретельно тестованому коді із застосуванням Agile методів. [9]

Можливості:

- Напевно всі компоненти орієнтовані на PHP 5 та E\_STRICT - сумісні
- Вбудований генератор коду
- Архітектура «використовуй тільки необхідно» з мінімальними залежностями компонентів
- Легкорозширюваний шаблон проектування MVC, підтримує макети і PHP-скрипти подання за замовчуванням
- Підтримує безліч різних баз даних, включаючи MariaDB, MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite, and Informix Dynamic Server
- Класи для створення, відправлення, отримання email з допомогою mbox, Maildir, POP3 та IMAP4
- Дуже гнучка система для кешування за допомогою безліч сховищ

Недоліки:

- Досить великий
- Повільний, бо без кешування
- Дуже складна архітектура, без глибокого розуміння шаблонів проектування та важкий для освоєння

- Слабке російськомовне співтовариство
- Повільна швидкість розвитку
- Останнє оновлення вимагає PHP 5.2
- Відсутній ORM.

**CakePHP** є фреймворком для PHP, який надає розширену архітектуру для розробки, обслуговування і розгортання web-додатків. Він використовує відомий шаблон проектування MVC, як і в об'єктно-реляційних фреймворках. Основною парадигмою CakePHP є збільшення продуктивності розробки, і, як наслідок, допомога програмісту у вигляді зменшенні обсягу написання коду. Спочатку створювався як клон популярного Ruby on Rails, тому багато ідей були запозичені саме звідти. [10]

Можливості:

- підтримка PHP4 (до версії 1.3 включно) і PHP5
- диспетчер URL, застосування регулярних виразів
- генерація коду за схемою бази даних (якщо дотримуватися стандарту іменування стовпців)
- перевірка форм
- компоненти для авторизації, обмеження доступу (ACL), управління сесіями, cookies, подання деревоподібної інформації (у вигляді Nested Sets)
- хелпери(компоненти) для генерації та заповнення форм, поділу на сторінки (paginate), управління кешем, JavaScript( AJAX також)
- гарні механізми інтернаціоналізації
- генерація SQL-запитів, також для таблиць з відносинами один до багатьох і багато до багатьох, ORM
- Scaffolding і генерація CRUD-сторінок для сутностей, Router::mapResources з Put Delete Get Post.
- Автогенератор коду Bake

- Міграції
- Консольна інтеграція, клас Shell і завдання Task
- Плагіни (як окремі програми), компоненти і поведінки
- Робота з Simple Test

Недоліки:

- Мала продуктивність
- Слабка документація
- Нестійкість до CSRF-атак
- Немає російськомовної спільноти
- Складний в розумінні ACL
- Дуже сурові угоди з іменування
- Низька швидкість розвитку.

**Ruby on Rails** - об'єктно-орієнтований програмний каркас (фреймворк) для створення веб-додатків, написаний на мові програмування Ruby. Ruby on Rails надає каркас модель-вид-контролер (Model-View-Controller) для веб-додатків, а також забезпечує їхню інтеграцію з веб-сервером і сервером бази даних. [11]

Можливості:

- Лаконічний синтаксис, зустрічається вплив Ада, Eiffel і Python
- Може обробляти винятки в стилі Java і Python
- Дозволяє переозначувати оператори методами
- Простий, легкий та послідовний синтаксис
- Можливість прямих системних викликів
- Швидка поява змін під час розробки
- Відсутність стадії компіляції
- Повністю об'єктно-орієнтована мова програмування. Всі дані в Ruby є об'єктами в розумінні Smalltalk. Має лише один виняток — керівні конструкції, які в Ruby, на відміну від Smalltalk, не є об'єктами

- Автоматичний прибиральник для сміття. Це працює для всіх об'єктів Ruby, в том реалізований додаванням потрібних методів одному конкретному об'єктові.
- Є можливість динамічно завантажувати розширення, якщо це дозволяє операційна система
- Перенесена на багато платформ. Мова розроблялася на GNU/Linux. Вона працює на багатьох версіях Unix, DOS, Microsoft Windows (частково, Win32), Mac OS, BeOS, OS/2 і т.д.

Недоліки:

- Мала швидкість роботи
- Проекти є досить громіздкі
- Складний debug коду
- Філософія розвитку багато в чому залежить від "магії" і "припущень"
- Жахлива російськомовна документація.

Apache Struts — це веб-фреймворк з відкритим кодом, який використовується для розробки Java EE вебзастосунків. також розширює Java Servlet API, надаючи архітектуру MVC (Модель-Вид-Контролер).

Можливості:

- Проста дії POJO-based
- Легке тестування коду
- Безпечність потоку передачі даних
- Підтримка AJAX
- Клієнтські технології AJAX валідації
- Можливість використовувати шаблони
- Підтримка різних типів результатів
- Дуже вільно розширюється за допомогою плагінів

- забезпечення роботи плагінів фреймворку Spring (впровадження залежностей)
- JFreechart плагін (діаграми)
- jQuery плагін (підтримка AJAX, UI віджетів, динамічні таблиці, діаграми)
- Rome плагін

Недоліки:

- Погана російськомовна документація
- Мале російськомовне співтовариство
- Складність для освоєння
- Важкість оптимізації додатків
- Складність для роботи із кількома розробниками

**ASP.Net MVC** - фреймворк для створення веб-додатків, який реалізує шаблон Model-view-controller. Даний фреймворк розроблений компанією Microsoft. [12]

Фреймворк ASP.NET MVC надає наступні можливості:

- Поділ завдань програми (логіка введення, бізнес-логіка і логіка інтерфейсу), надає широкі можливості для тестування і розробки на його основі. Всі контракти платформи MVC створені на інтерфейсі і потребують тестування за допомогою макетів об'єкта, які імітують поведінку реальних об'єктів програми. Додаток можливо використовувати на модульному тестуванню без запуску контролерів у процесі ASP.NET-це прискорює процес і робить його більш гнучким. Для тестування потрібно використовувати будь-які платформи, які є сумісними з .NET Framework.

- Платформа надає можливість для розширення, додавання, доповнення . Компоненти платформи ASP.NET MVC можна дуже просто замінити або налаштувати. Розробник проекту має можливість підключати

свій механізм уявлень, політику маршрутизації URL-адрес, серіалізацію параметрів методів, дій та інші компоненти. Платформа ASP.NET MVC також підтримує використання моделей контейнера впровадження залежності (DI), інверсії елемента керування (IOC). Також модель для впровадження залежності має змоги впроваджувати об'єкти в клас, не очікувати на створення об'єкта самим класом. Модель інверсії елемента управління вказує на те, що об'єкт вимагає інший об'єкт, тоді, звісно, перші об'єкти мають отримати другий об'єкт застосовуючи зовнішнє джерело (наприклад, з файлу конфігурації), що полегшує тестування.

- Наявна широка підтримка маршрутизації ASP.NET. Він є потужним компонентом у зіставленні URL-адрес та дозволяє створювати додатки із зрозумілими URL-адресами, що можна використовувати в пошуку. URL-адреси не повинні включати розширення імен файлів і призначені для підтримки шаблонів іменування URL-адрес, які забезпечують адресацію, оптимізовану для пошукових систем (SEO) і для передачі репрезентативного стану (REST).

- Створена підтримка у використанні розмітки в існуючих файлах сторінок ASP.NET (ASPX), елементів управління (ASCX) і головних сторінок (MASTER) як шаблонів представлень. Разом з платформою ASP.NET MVC можна користуватися і існуючими функціями ASP.NET. Наприклад, вкладені головні сторінки, вбудовані вирази, декларативні серверні елементи управління, шаблони, прив'язку даних, локалізацію і т. д. Недоліки:

- Високий поріг входження
- Немає механізму зберігання стану
- Важкість створення бібліотек компонентів.

Кожен зі згаданих фреймворків має свої переваги та недоліки, тому їх вибір необхідно здійснювати з урахуванням проекту, який буде

створюватися. Це є головним критерієм вибору, адже жоден із фреймворків не є універсальним рішенням для будь-якого проекту. Далі наведена таблиця 1.2 порівняння основного функціоналу вищезгаданих фреймворків

Фреймворк							
Критерій	Zend	CakePHP	Symfony2	Yii	ASP.NET	ROR	Struts
Мова	PHP	PHP	PHP	PHP	ASP.NET	Ruby	Java
Шаблон проектування MVC	+	+	+	+	+	+	+
Застосування ORM	+	+	+	+	ORM– незалежна	+	+
Інтеграція з БД	+	+	+	+	***	+	***
Механізми інтернаціоналізації і локалізації	+	+	+	+	***	+	+
Використання шаблонів при створенні інтерфейсів користувача	+	+	***	+	+	+	+
Створення і перевірка форм	+	+	+	+	+	+	+
Управління доступом на основі ролей	+	+	***	+	+	+	ні
Використання AJAX	+	+	***	+	+	+	+
ЧПУ (Friendly URL)	+	+	+	+	+	+	ні
Модульне тестування (unit–testing)	+	+	***	+	+	+	+
Система кешування	+	+	+	+	+	+	ні

Таблиця 1.2 - Порівняльні характеристик

#### 1.5.5. Загальні переваги використання Back-end фреймворку

Під час вибору фреймворку виникають деякі труднощі, які пов'язані з визначенням завдань, які він може виконувати, та його призначення. Як бачимо, не всі фреймворки підтримують ORM для роботи з базами даних, мають якісну інтеграцію і гарну документацію. Якщо для створення нового сайту треба знайти зручний і комфортний у використанні фреймворк, то треба ретельно підійти до питання його вибору та зважити всі плюси і мінуси.

Розглянемо загальні переваги використання фреймворку:

- легкість у розробці та розвитку проекту;
- ефективне використання ресурсів сервера;
- відкритий код;
- надійність web-розробок. фреймворк будується з базових, перевірених, налагоджених функцій і операцій. Він побудований на базі об'єктно-орієнтованого програмування;

- розвиток і вдосконалення фреймворка;
- велика різноманітність супровідної документації та прикладів розробки на різних мовах;
- популярність, велика кількість розробників, співтовариств;
- легкість супроводу проекту в майбутньому, тому розробка із застосуванням фреймворка заснована на певних угодах;
- фреймворк дозволяє сконцентруватися на вирішенні архітектурних завдань, а не базових, як при розробці без його застосування;
- фреймворк дозволяє вузько вирішувати поставлену задачу.

Однією з основних переваг фреймворка являється зручна розробка нестандартних проектів. Не один великий нестандартний проект (наприклад, twitter.com, фотобанк з онлайн купівлею фотографій, сайт знайомств і т.д.) не побудований на готовій CMS – вони для цього не призначені. Усі цікаві та нові проекти розробляють на фреймворках. Web-проект, розроблений за допомогою фреймворку, розвивається динамічно та швидко. За зміною вимог змінюється і сайт, достатньо тільки замінити окремий блок (модуль) та створити новий розділ, внести нововведення в дизайн.

#### **1.5.6. Недоліки застосування Back-end фреймворку**

Недоліки застосування фреймворку умовні і незначні порівняно з перевагами:

- 1 файл = 1 клас;
- багато коду не використовується, не лежить мертвим вантажем в проєкті;
- складність для освоєння;



- готових модулів і компонентів, які міг би встановити клієнт, в мережі Інтернет немає, навіть платних. Всі доробки та введення необхідно замовляти у розробників;
- розробка функціоналу, стандартного для сайтів на CMS, обійдеться дорожче і займе більше часу.

### **1.5.7. Web-сервер Apache**

Apache є кросплатформним програмним забезпеченням підтримуючи такими ОС як Linux, BSD, Mac OS, Microsoft Windows.

Головними достоїнствами Apache, вважаються надійність і гнучкість конфігурації. Він помагає підключати зовнішні модулі для надання даних, які використовуються СУБД для аутентифікації користувачів та для того, щоб модифікувати повідомлення про помилки.

Ядро Apache включає в себе основні функціональні можливості, такі як обробка конфігураційних файлів, протокол HTTP і система завантаження модулів. Ядро (на відміну від модулів) повністю розробляється Apache Software Foundation, без участі сторонніх програмістів [13].

Теоретично, ядро apache функціонує в чистому вигляді, без використання модулів. Але, варто зауважити, що функціональність такого рішення вкрай обмежена.

Ядро Apache написано на мові програмування C.

Система конфігурації Apache ще заснована на текстових конфігураційних файлах, а також має три умовних рівня конфігурації:

1. Конфігурація сервера (httpd.conf).
2. Конфігурація віртуального хоста (httpd.conf с версії 2.2 extra / httpd-vhosts.conf).
3. Конфігурація рівня директорії (.htaccess).

Має власну мову конфігураційних файлів, які засновані на блоках директив. Більшість параметрів ядра можуть бути змінені через конфігураційні файли, аж до управління MPM. Велика частина модулів має зовсім власні параметри.

Також Apache має вбудований механізм віртуальних хостів. Він дозволяє обслуговувати на одному IP-адресу безліч сайтів (доменних імен), відображаючи для кожного з них власне вміст. Існує безліч модулів, що додають до Apache підтримку різних мов програмування і систем розробки.

До них відносяться:

1. PHP (mod\_php).
2. Python (mod\_python).
3. Ruby (apache-ruby).
4. Perl (mod\_perl).
5. ASP (apache-asp).

Крім того, Apache підтримує механізми CGI і FastCGI, що дозволяє виконувати програми на практично всіх мовах програмування, в тому числі C, C++, sh, Java.

### **1.5.8. Шляхи та області використання**

Для того, щоб скористатися всіма можливостями фреймворка потрібен чималий багаж знань в розробці додатків. РНР-фреймворки допомагають усунути дуже часту помилку при програмуванні додатків, а саме - проблему повторення коду, систематизувати процес розробки. Фреймворк - це потужний інструмент для такої мови програмування, що швидко розвивається, як РНР, який допоможе організувати ваш код.

Кожна людина має різні вподобання і потреби. Для одного розробника використання фреймворків може допомогти у прискоренні процесу програмування, а для іншого це може здатися марною тратою часу. У більшості

випадків це залежить від рівня професіоналізму, але, в цілому, фреймворки призначені для того, щоб заощадити час і абстрагуватися від рутинних завдань.

В основному, фреймворки застосовуються для розробки проектів складніше, ніж дво- або тристорінковий сайт з текстовими сторінками.

## **1.6 Обґрунтування вибраного веб-редактора**

Зараз існує безліч програм для створення сайту та розміщення його в інтернеті. При цьому варто пам'ятати, що це лише інструмент, за допомогою якого розробник працює. Це не впливає на те, чи сайт буде привертати сайт увагу користувачів, чи буде він успішний у бізнесі. Це залежить тільки від маркетингу та фантазії дизайнера (розробника).

Для створення та редагування сайту було обрано Visual Studio Code. Переваги Visual Studio Code:

1. Безкоштовний редактор, вихідного коду.
2. Можливість віддаленого підключення до FTP.
3. Підтримка плагінів.
4. Автозбереження коду.
5. Автодоповнення коду які фіналізують, класи, методи, теги і т.д.

## **1.7. Висновок до розділу 1**

У даному розділі описане предметне середовище, за допомогою якого розробляється програмне забезпечення. Наведені основні поняття та визначення відповідно напрямку діяльності. Описаний процес діяльності. Розглянута та описана функціональна модель комплексу, виявлені головні дійові особи (актори), для детального опису моделі наведена схема структурна варіантів використання. Сформульоване призначення комплексу задач, а також визначені цілі та задачі комплексу задач та проведений аналіз з іншими сайтами кафедр.

## **2. ВИБІР ФРЕЙМВОРКУ ДЛЯ РЕАЛІЗАЦІЇ. ОГЛЯД РОБОТИ З ОБРАНИМ ФРЕЙМВОРКОМ. SEO-ОПТИМІЗАЦІЯ САЙТУ**

В даному розділі буде здійснено вибір фреймворку. Після вибору буде розкрито особливості обраного фреймворку та принципи створення web – додатків за його допомогою. Також розглянемо SEO-оптимізацію, та реалізуємо на сайті.

### **2.1 Вибір Front-end фреймворку для реалізації сайту**

Twitter Bootstrap це неперевершений фреймворк для створення сучасних, кросбраузерних і стандартизованих інтерфейсів для розробників. Продумана структура коду HTML, JavaScript і CSS дає вам можливість створювати безліч самих різноманітних елементів інтерфейсу і сітку сайту.

Свідченням того, що Bootstrap це дуже вдала розробка, є його безперервно зростаюча популярність. Нинішній рівень розвитку цього фреймворку вже дозволяє повністю створити будь-який web-інтерфейс.

#### **2.1.1 Bootstrap структура**

Блокова система (Grid System) Bootstrap допомагає горизонтально розділити сторінку на 12 стовпців. Використовувати 12 частин немає необхідності, але ці блоки можна легко групувати, створюючи більш широкі стовпці. Вони будуть складатися з двох, трьох і так далі частин сторінки, розділеної на 12 частин (Рисунок 2.1) .

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Рисунок 2.1 – блоки Bootstrap

Блокова структура в Bootstrap розрахована також на мобільні пристрої та планшети. Блоки будуть автоматично перебудовуватися в залежності від розміру екрана.

Базова структура макета за допомогою блокової системи Bootstrap. (Рисунок 2.2)

```

<div class="row">
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
</div>
<div class="row">
  <div class="col-md-8">.col-md-8</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-6">.col-md-6</div>
</div>

```

Рисунок 2.2 – структура макета

### 2.1.2 Класи для комірок

У блокової верстці Bootstrap є тільки чотири класи:

- xs (extra small) - для мобільних телефонів, до 767 пікс. Включно
- sm (small) - для планшетів, розмір екрану 768-991 пікс.
- md (middle) - для десктопів, 992-1199 пікс.
- lg (large) - для великих екранів, від 1200 пікс.

Принадність верстки на Bootstrap в тому, що ці класи ви можете комбінувати, щоб створювати адаптивну верстку і точно знати, як макет буде відображатися на тому чи іншому розмірі екрану.

Призначення певних аспектів системи сітки Bootstrap для різних пристроїв (Рисунок 2.3).

	Дуже вузькі пристрої Телефони (<768px)	Вузькі пристрої Планшети (≥768px)	Середні пристрої Монітори (≥992px)	Широкі пристрої Монітори (≥1200px)
Поведінка сітки	Горизонтальна завжди	Складена спочатку, але горизонтальна, коли ширша за контрольні точки		
Ширина контейнера	Немає (автоматична)	750px	970px	1170px
Префікс класа	.col-xs-	.col-sm-	.col-md-	.col-lg-
# колонок	12			
Ширина колонки	Автоматична	~62px	~81px	~97px
Ширина проміжку	30px (15px з кожної сторони колонки)			
Можуть вкладатись	Так			
Можуть зміщуватись	Так			
Порядок колонок	Так			

Рисунок 2.3 – опції сітки Bootstrap

#### 2.2.1. Процес розробки сайту з використанням Bootstrap

Розглянувши фундаментальні концепції Bootstrap, треба описати загальний процес створення веб-додатків, якщо використовувати фреймворк.

Основні етапи при розробці додатку виглядають так:

1. Завантажуємо бібліотеку Bootstrap, розпакуємо файли в корінь сайту.
2. Підключаєм Bootstrap до сайту.
3. Створюємо таблиці стилів, які містять базову CSS-розмітку напевно для кожного HTML-елемента, що забезпечує їм однаково привабливий вигляд.
4. Розмітка сторінок сайту.
5. Медіа-запити, для адаптивного дизайну сайту.

## **2.2 SEO-ОПТИМІЗАЦІЯ САЙТУ**

SEO - це оптимізація сайту під алгоритми пошукових систем з тією метою, щоб сторінки відображалися в результатах видачі по тим запитам, які вам вигідні в плані залучення цільової аудиторії. Причому кінцева суть SEO полягає в тому, щоб висунути ці самі сторінки якомога ближче до верхніх результатами. До цього можна прийти по-різному, а тому ви повинні знати, які існують види SEO-оптимізації сайтів, і які з них слід використовувати. [14]

### **2.2.1. Види SEO-оптимізації**

Білий вид SEO-оптимізації сайту - це дуже чесний шлях, за який важко борються пошукові системи. В цьому випадку просування рухається природним шляхом: обмінюється бек-лінками з майданчиками такої ж тематики, ресурс часто оновлюється і поповнюється, на нього з'являються посилання на форумах, блогах. Жодної спроби «зламати» пошуковик і використовувати його алгоритми в корисливих цілях не робиться. Цей метод дуже довгий, однак його застосування веде за собою не менш довгострокові перспективи розвитку в пошукових системах.

Сірий вид SEO-оптимізації сайту в свою чергу вже не передбачає настільки «чистих» намірів, але на офіційному рівні він не забороняється. На сьогоднішній день актуальні такі дії як: покупка посилань, використання дорвеев, але з

відмовою від переадресації, накрутка лічильників відвідуваності для зростання позицій в рейтингах, але при роботі з даним видом SEO-оптимізації сайту головне не переборщити, тому що з однаковим успіхом ви можете як накрутити позиції ресурсу, так і опустити їх. В цьому відношенні слід діяти дуже обережно, щоб не наштовхнути пошукачів на зайві підозри.

Помаранчева SEO-оптимізація це заволодіння трафіку не цілком раціональними шляхами. В даному випадку може бути застосована стратегія, спрямована на залучення будь-яких відвідувачів, навіть не цільових. Реалізується це шляхом додавання ключових фраз, мало що відносяться по темі до текстового вмісту, але користуються великим попитом на в рядах онлайновцев. Насправді, не цілком виправданий метод, бо навіщо тоді весь цей сир-бор, якщо в результаті на вашому проекті буде затримуватися тільки мінімальна частина від всього трафіку ?!

Чорний вид SEO-оптимізації сайту на зараз повністю вичерпав себе. Раніше він будувався на різних хитрощах, здатних обдурити пошукові системи і вийти в ТОП підпільними шляхами. Сьогодні пошуковики бачать таку тактику наскрізь, так що з цим ви точно отримаєте швидше бан, ніж якісь позитивні підсумки.

### **2.2.2. Методи SEO**

Сьогодні актуальна внутрішня, так і зовнішня оптимізація сайтів, причому найкращі результати може принести тільки комплексне використання цих методів.

Внутрішня оптимізація сайту - зміна вмісту ресурсу із метою повної і більш швидкої його індексації пошуковими системами. це:

- Розміщення нового контенту та редагування старого.
- Створення файлу robots.txt або його коригування.
- Редагування мета-тегів description і title.
- Редагування програмного коду.



- Внутрішня перелінковка сторінок.
- Робота з ключовими словами.
- Зміна юзабіліті.
- Зміна дизайну.

Зовнішня оптимізація сайту - просування ресурсу за допомогою сторонніх сил і засобів. це:

- Розміщення прес-релізів, відгуків, новин та іншого цікавого контенту на сторонніх майданчиках.
- Усунення різноманітних недоліків, що можуть привести до падіння позицій і блокування ресурсу пошуковими системами.
- Постійний моніторинг позицій сайту в пошукових ведучих.
- Реєстрація сайту в каталогах і пошукових системах.
- Використання соціальних мереж і блогів.
- Обмін посиланнями з іншими ресурсами.
- Придбання вічних посилань.

Це точно не повний перелік робіт по внутрішній і зовнішній оптимізації сайту. Він постійно доповнюється і змінюється, з'являються все нові методи просування, причому деякі з них відносяться до недозволених.

### **2.2.3. Мета-теги**

Мета-теги від англійського слова "meta-tags" - це HTML теги, призначені для обробки пошуковими системами сторінок веб-сайту. Інакше, мета-теги - це помічники пошукових систем, які являють собою зміст про певну веб-сторінку[15].

Мета-теги спрощують, конкретизують пошук нової інформації на сайті. Їх можна вставляти в кожен сторінку сайту, коротко описуючи її зміст з допомогою ключів і тегів. Також розміщувати мета-теги, які не копаються в код сторінки можна за допомогою плагіна «SEO Ultimate». Мета-теги складаються з певних умовних операторів, які поділяються на:

- Description - Опис сторінки;
- Keywords - Ключові слова;
- Robots - Дозвіл на індексування сторінок;
- Revisit-After - Час і інтервал відвідування пошукового робота;
- Author - Ім'я автора або Компанії

#### **2.2.4. Вимоги до SEO оптимізації**

Після створення web-сайту вкрай необхідно змусити його працювати на себе - зробити його відомим ової аудиторії. Інакше це будуть викинуті на вітер гроші. Коротко перерахуємо основні інструменти Internet-маркетингу, з допомогою яких можна вирішувати цю задачу.

- Розробка і оптимізація SEO компонентів сторінок сайту на моменті проектування сайту;
- Розміщення (індексування) сайту в пошукових системах, каталогах і рейтингах, таких як Google, Rambler, Yahoo;
- Реклама на тематичних і відвідуваних сайтах;
- Контекстна реклама;
- Участь в банерообмінних мережах.

Розробивши та опублікував сайт в інтернеті, його корисно зареєструвати в пошукових системах, рейтингах, каталогах. Це дозволить легко виявити його користувачам Internet, хто цікавиться сайтами даної тематики а також дозволить вам стежити за відвідуваністю вашого сайту і мати додаткову статистичну інформацію про його відвідувачів і відвідини.

Реклама на популярних сайтах (google.com, rambler, yahoo) - найбільш ефективна, а форми такої реклами на сайті можливі самі різні: банери і кнопки на окремих або на всіх сторінках, текстова і банерна реклама в розсилці, Web-сторінка в Web-каталозі (це дозволить мати своє представництво в Internet на відвідуваному сайті при мінімальних фінансових витратах), і ін. Корисну для

відвідувачів (Потенційних клієнтів) інформацію можна розмістити за підписом фірми в різних розділах і сприяти позитивному іміджу фірми.

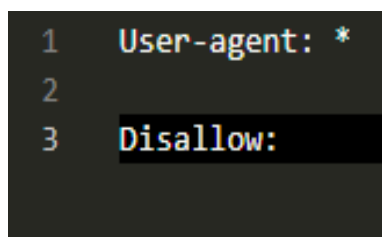
Це приклад можливостей для представників бізнесу. Подібні проекти і можливості існують для багатьох напрямків бізнесу. І не треба забувати, що аудиторія Internet відрізняється освіченістю і кваліфікацією, тому в ній велика частка людей, які називаються "лідерами думок", до яких часто звертаються за інформацією або радою (наприклад, купівля будівельних матеріалів, покупки комп'ютера, електроінструменту та ін.).

Сьогодні далеко не для всіх напрямків діяльності Web-сайту може окупати витрати на його створення і підтримку. З цього необхідно максимально ефективно провести роботу по SEO оптимізації нашого вебсайта. Даний етап значно скоротить витрати на подальше просування ресурсу в мережі інтернет.

Індексація сайту - це обхід роботами пошукових систем сайту, і внесення його в базу. Пошукові роботи заносять в базу інформацію про сайт і всіх його сторінках, які доступні для пошуку. Пошукової бот індексує посилання, зображення, відео, і інші елементи на сайті.

Покращення індексації сайту кафедри. Даний додаток має 549 google-index. Створення індексації в 5 етапів:

- Створення файлу robots.txt (рисунок 2.4)



```
1 User-agent: *
2
3 Disallow:
```

Рисунок 2.4 – robots.txt

Також за допомогою рядка Disallow ми закриваємо непотрібні сторінки, в нашому випадку це службові розділи, або ті розділи, що не потрібно індексувати.

- Додати сайт в пошукові системи

Додати сайт в Google webmaster tools

- Поставити посилання з інших сайтів
- Посилання з соціальних мереж
- Додавання сайту в соціальні закладки

За допомогою соціальних закладок можна швидко залучити пошукових роботів.

Для того щоб максимально швидко проіндексувати сайт, найкраще використовувати всі методи відразу, тоді і віддача буде краще.

Якщо сайт довго не індексується, потрібно перевірити в першу чергу, не закрито він від індексації, якщо все нормально, то вищеописаними методами залучити пошукових роботів на сайт.

### **2.2.5 Додатки для оптимізації**

Для оптимізації сайтів є багато різних додатків для розширення, але найкращі це від компанії Google та Yandex. Саме ці великі компанії є кращими, бо вони мають свою пошукову систему. Google та Yandex мають веб майстерні. Перевірку оптимізації роблять також схоже, вони перевіряють в першу чергу:

- Метатеги до сторінок;

Метатеги title і deskription - це елементи в HTML-кодi сторінки, які описують її назву і вміст. Їх наявність необхідно для коректної пошукової видачі. Важливо, тоді вони були унікальними на кожній індексується сторінці сайту. Мета теги впливають не тільки на ранжування, а й кількість переходів з видачі, так як описують користувачеві контент ще до того, як він клікнув на посилання.

- Ключові слова;

Це основний метод, який необхідний для просування. Ключі - це слова передбаченням вимог відповіді на які користувач отримає на вашому ресурсі. Вони містяться в контенті, дають пошуковим системам зрозуміти кому показувати ваш сайт і наскільки релевантно вміст сторінок призначеним для користувача запитів.

Якщо на в сайті йдеться про послуги автомайстерні, то навіть при великому бажанні вам не вдасться уникнути слів «ремонт» або «автомобіль». Але щоб брати участь у видачі по всіх можливих запитах цієї тематики потрібно скласти повне семантичне ядро і грамотно розподілити його по сторінках.

Додаючи в текст ключі, також дуже важливо не перестаратися з їх кількістю. Якщо частота слова по [сервісу advego.com/text/seo](http://сервісу advego.com/text/seo) перевищує 3.5%, то текст можна вважати заспамленості, а значить він буде гірше ранжируватися.

- Контент;

Контент повинен не тільки містити ключові слова, але і бути унікальним та особливим. А тому просто скопіювати його з інших джерел не можна. Створювати унікальне текстове наповнення можна самостійно, або скориставшись послугами копірайтерів. Перевірити статтю на плагіат можна в [сервісі text.ru](http://сервісі text.ru).

Треба враховувати водність (кількість слів, що не несуть смислового навантаження) і тошнотно (повторювані слова або фрази). Нормальний показник водності вважається не вище 60%, класична нудота від 3% до 5%, академічна - не вище 9%. Ще один критерій ранжирювання - це наявність заголовка H1 і підзаголовків H2-H6. Вони полегшують візуальний пошук інформації по тексту і прискорюють процес індексації.

- Структуру;

Структура сайту впливає відразу на кілька факторів ранжирювання. А тому і вимоги до неї високі.

В першу чергу структуру доповнюють і розширюють під семантичне ядро: створюють окремі сторінки під кожен групу запитів. Для зручної навігації, посадочні сторінки групуються в системи категорій і підкатегорій. Після чого організовують їх перелинковку таким чином, щоб на кожен зі сторінок можна було потрапити з головної не менш ніж за три переходи.

Але не можна обтяжувати сайт численними внутрішніми посиланнями, інакше пошуковий бот не зможе проіндексувати його до кінця. Тому в залежності від розміру ресурсу застосовують різні види структур, і закривають зайві переходи від індексації.

- Швидкість завантаження сторінок;

Тривалість завантаження - один важливий фактор, який впливає на поведінкові показники. Нормальне час - не довше 2 секунд. Повільні ресурси і ранжуються нижче, і гірше конвертують відвідувачів.

- Технічні фактори;

Також до завдань внутрішньої оптимізації входить моніторинг та усунення 404 помилок, дублів сторінок і битих посилань. Якщо подібних помилок на сайті занадто багато, вони також можуть знизити позиції.

До зовнішнього просування треба приступати після внутрішньої оптимізації. Інакше існуючі технічні помилки можуть перешкодити бажаного росту проекту, висмоктавши бюджет на посилання даремно.

Google має пару додатків для оптимізації сайтів:

- Google Analytics;
- Google Tag Manager;

Tag manager - рішення від Google, яке дозволяє розміщувати скрипти на сторінках сайту без безпосереднього втручання в його код. GTM дає можливість додавати і оновлювати скрипти безпосередньо через web-інтерфейс, а також задавати правила активації цих скриптів. Диспетчер тегів Google - це простий, надійний і безкоштовний інструмент. Маркетологам він забезпечує гнучкість в роботі, а веб-майстрам дозволяє зосередитися на найважливіших завданнях, не турбуючись про дрібниці. Це рішення для фахівців з маркетингу, в якому доступний єдиний інтерфейс управління всіма тегами на сайті. [16]

- Google Optimize;

Google Optimize - це зручний інструмент, який дозволяє проводити А / В і різноманітні тестування і збирати дані про поведінку користувачів на сайті.

За допомогою цього інструменту фахівець створює копію тестируемой сторінки, але вносить деякі змінами в її структуру або елементи. Наприклад, можна змінити розташування блоків на сторінці, змінити колір або розмір кнопки, замінити зображення. Адже зовнішній вигляд сайту значно впливає на поведінку користувачів і відповідно на коефіцієнт конверсії сайту.[17]

У звітах Optimize показує докладне порівняння варіантів і допомагає визначити, який з них краще справляється з поставленими завданнями і цілями.

А / В тестування не є чимось особливим, бо як і надання персоналізованого контенту на основі поведінки клієнта. Головна перевага полягає в тому, що Google Optimize працює в парі з Google Analytics.

Веб – майстерня Yandex не розглядаємо для оптимізації сайту, так як на території України заблокована компанія Yandex.

## **2.3 Висновки розділу 2**

В даному розділі було розглянуто які є фреймворки, та було обрано для створення проекту фреймворк Bootstrap. Вибір зробили на фреймворкі Bootstrap, так як має гнучкість, бібліотека важить не багато, але має багатий функціонал, також посібник багато мов, як українська, російська, англійська та інші.

Для оптимізації додатку використали розширення, такі як Google Analytics, Google Tag Manager, Google Optimize.

### 3. УДОСКОНАЛЕННЯ САЙТУ ЗА ДОПОМОГОЮ BOOTSTRAP ТА SEO ОПТИМІЗАЦІЯ

В даному розділі буде розглянуто удосконалення сайту для кафедри ПСОН за допомогою використання WEB-фреймворку Bootstrap. Буде описано етапи удосконалення сайту за допомогою фреймворку, на сервері кафедри, а також оптимізація за допомогою Google.

#### 3.1. Загальна структура сайту

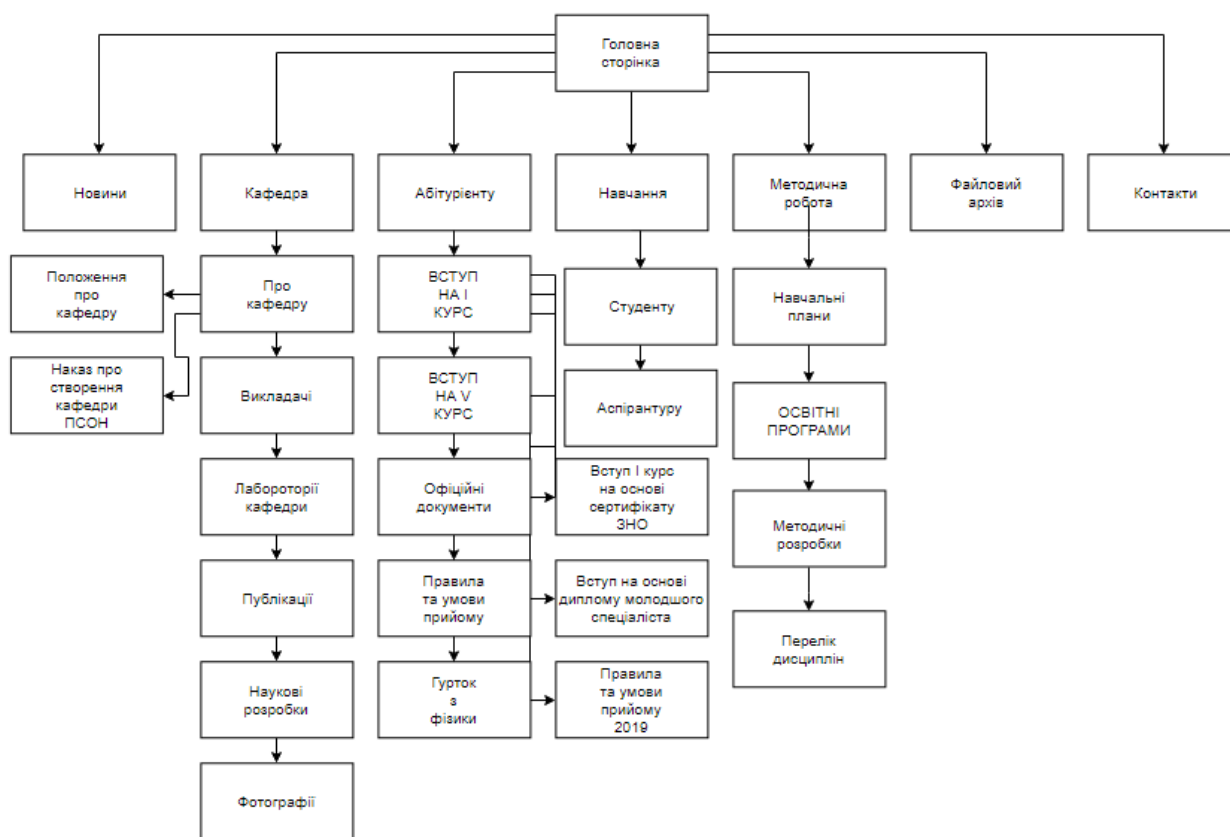


Рисунок 3.1. - Структурна схема сайту

На рисунку 3.1. представлена структурна схема сайту, що враховує всю специфіку, яка розробляється.

На головній сторінці тепер відображаються останні новини, наукові розробки і основна інформація.



Сторінка «Про кафедру» містить додаткові вкладки «Положення про кафедру» «та наказ про створення кафедри ПСОН». Тут відображається історія кафедри з моменту її виникнення і до цього дня.

Сторінка «Студенту» містить інформацію для студентів, літературу, файловий архів.

Сторінка «Абітурієнту» містить всю необхідну абітурієнтам інформацію про спеціальності.

Сторінка «Новини» містить останні оновлення.

Сторінка «Викладачі» всі викладачі кафедри.

Сторінка «Контакти» містить інформацію про те як зв'язатися з представниками кафедри.

Сторінка «Аспірантура» містить інформацію про аспірантів, також має інформацію для аспірантів.

Сторінка «Публікації» наукові публікації кафедри.

Сторінка «Методична робота» навчальні та робочі плани.

Сторінка «Наукові розробки» наукові розробки кафедри.

Сторінка «Конференція» конференція приладобудування.

Сторінка «Фотографії» фотографії кафедри ПСОН.

Сторінка «Файловий архів» містить навчально-методичні розробки, курсові та дипломні проектування.

### **3.2. Удосконалення інтерфейса**

Для удосконалення використовували фреймворк Bootstrap, спочатку треба скачати фреймворк з офіційного сайту, файл, що скачали потрібно розпакувати в корінь сайту та підключити до сайту. Підключаємо фреймворк через `<Link>` - встановлює зв'язок із зовнішнім документом. Так же підключимо бібліотеку JQuery, але уже по іншому рисунок 3.1.

```
<script type="text/javascript" src="jquery.js"></script>
```

Рисунок 3.1 – підключена бібліотеку JQuery

До удосконалення сайту цієї кафедри мав такий вигляд рисунок 3.2.

Даний сайт був створений без допомоги фреймворків та сторонніх бібліотек, також він немає гнучкості.

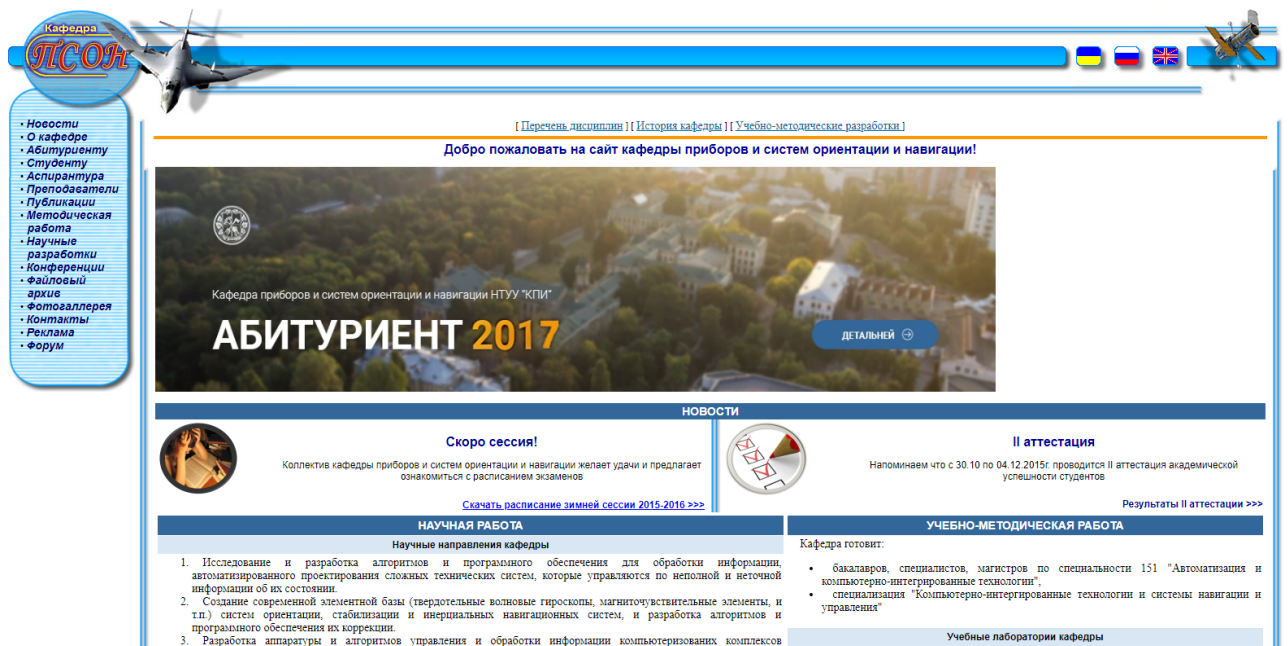


Рисунок 3.2 – головна сторінка сайту до удосконалення

За допомогою bootstrap створили адаптивну головну сторінку сайту, а за допомогою JQuery зробили шапку сайту динамічною. Дизайн та структура коду змінилась кардинально рисунок 3.3.

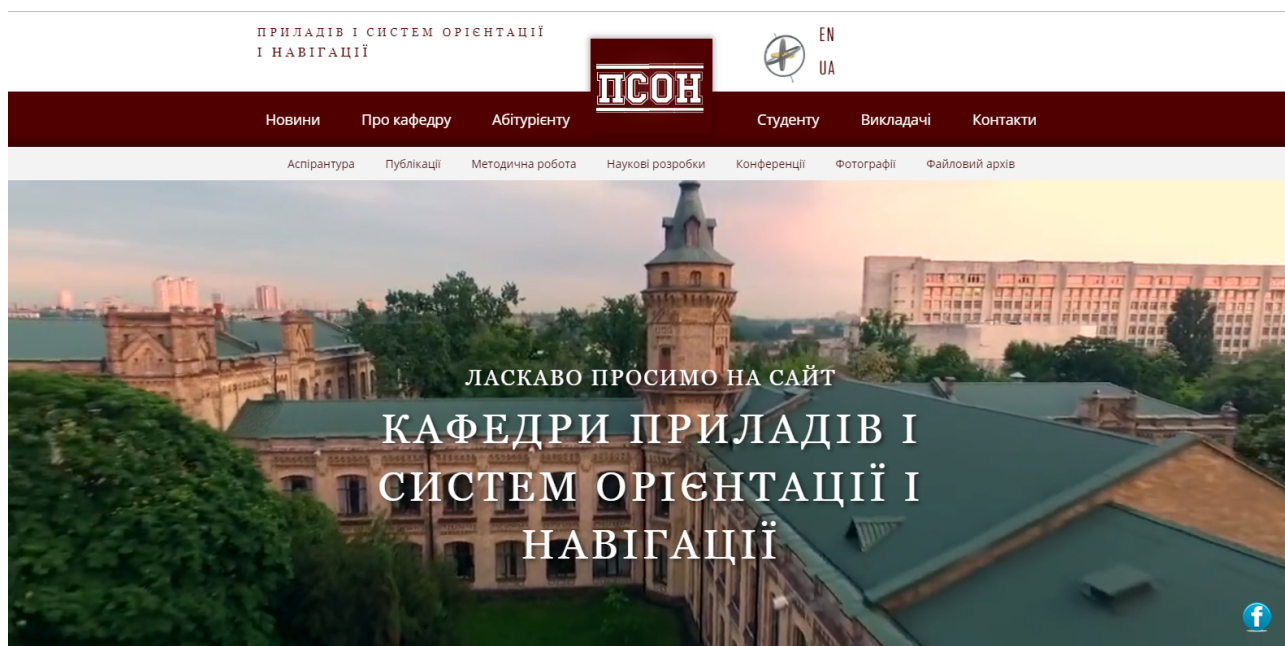


Рисунок 3.3 – головна сторінка сайту після удосконалення

Раніше на сайті використовували HTML 2.0, на даний момент використовуємо HTML 5.0, це більш удосконалена мова розмітки.

### 3.3 Вдосконалення SEO-оптимізації сайту

Для того щоб, сайт могли знайти в пошукових системах потрібно SEO-оптимізації сайту. Оптимізацію виконали завдяки додаткам від Google. Спершу було створено файл robots.txt, цей файл потрібен, для пошукових ботів. Чим більше ботів потраплять на сайт, тим більший індекс отримаємо.

Файл завантажуюмо в корінь сайту, та в мета тегу вказуємо, що на сайті є файл рисунок 3.4.

```
<meta name="robots" content="noindex, follow">
```

Рисунок 3.4 – мета тег robots

Щоб пошукові боти точно змогли орієнтуватися по сайту треба створити мапу, для цього використали генератор мап xml-sitemaps.com, створений файл потрібно завантажити в корінь сайту.

До удосконалення сайту індекс становив 549, після того як створили файл robots.txt, за один день індекс виріс до 681. Але це малий результат, щоб покращити потрібно прописати мета-теги, такі як description та keywords рисунок 3.5.

```
<meta name="keywords" content="КПІ, НТУУ, навчання, наука, інноватика, політехніка,">
<meta name="description" content="КПІ, НТУУ, навчання, наука, інноватика, політехніка,">
```

Рисунок 3.4 – мета тег description та keywords

Аналізувати теги використовуємо сайт tagmanager.google.com рисунок 3.5. «Диспетчер тегів» - система управління тегами, яка створена Google для управління тегами JavaScript і HTML, які використовуються для відстеження та аналізу веб-сайту.

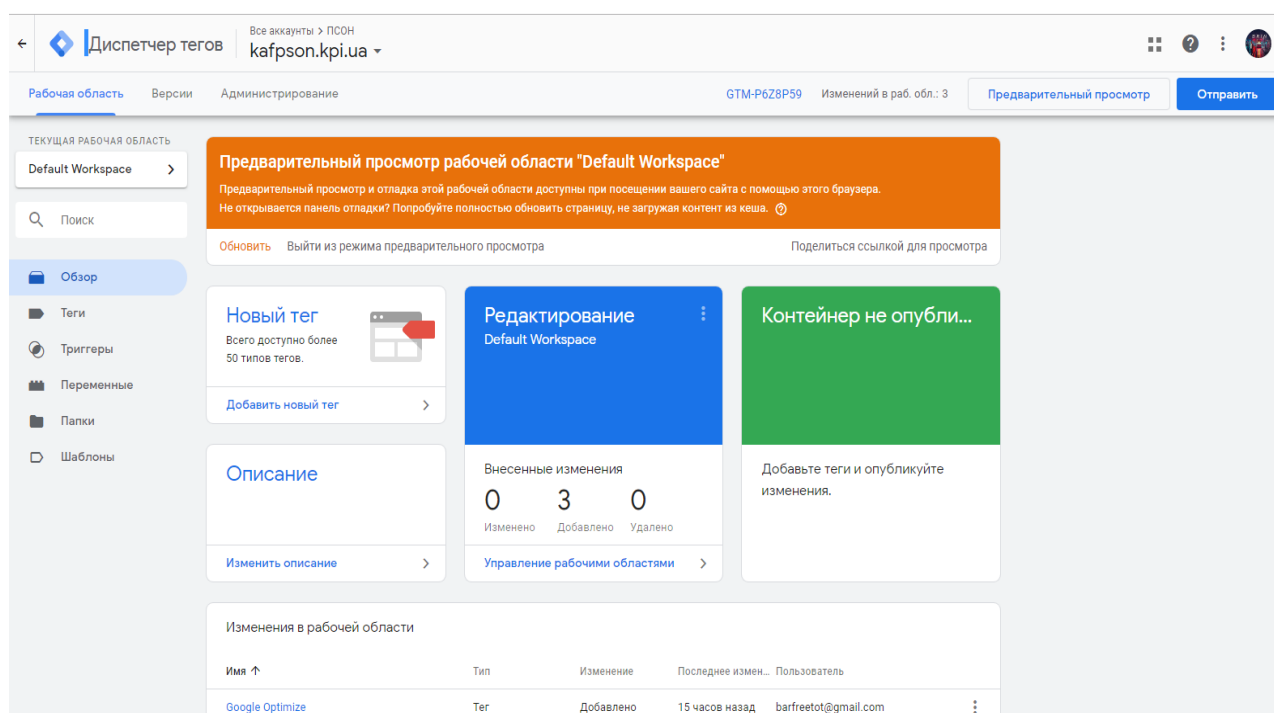


Рисунок 3.5 – Диспетчер тегів

При налаштуванні диспетчер тегів потрібно додати JS-код в `<head>` `</head>` та завантажити розширення для браузера, потім на сайті можемо аналізувати (рисунок 3.6)

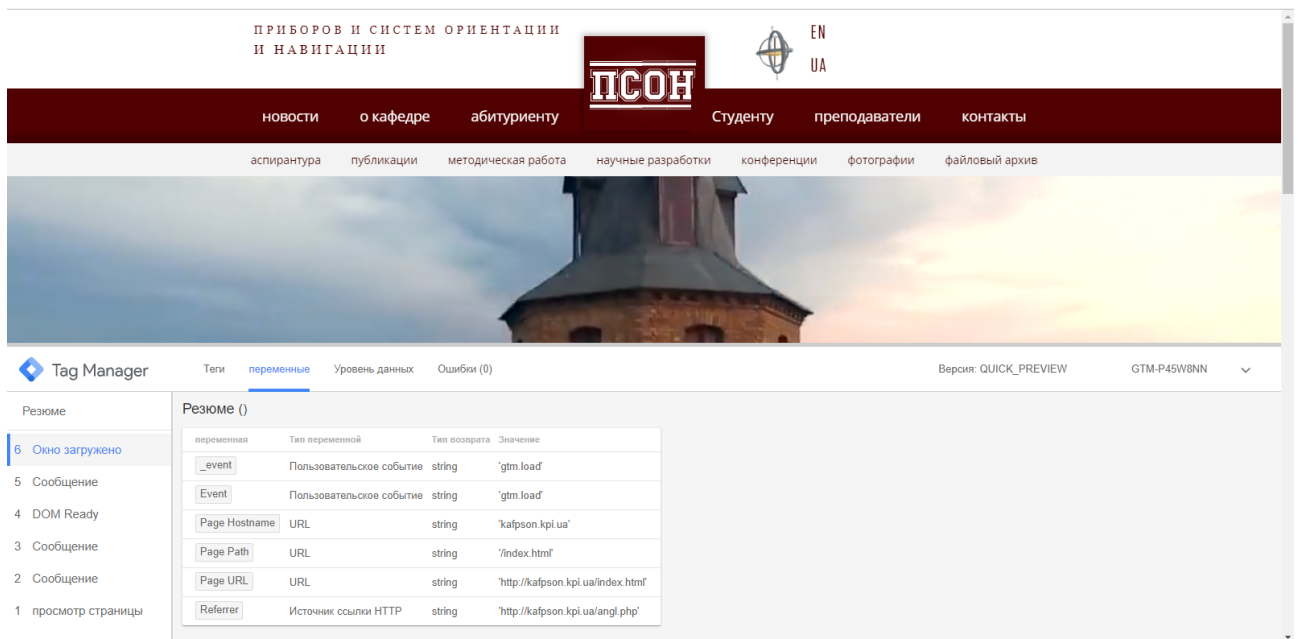


Рисунок 3.6 – розширення для браузера «Tag Manager»

Для того, щоб знати скільки людей відвідали сайт, з якої країни, з якого пристрою, використовували додаток Google Аналітика.

Google Analytics - це безкоштовний сервіс, що надається Google для створення детальної статистики відвідувачів веб-сайтів. Статистика збирається на сервері Google. Для роботи додатка, потрібно додати JS-код в `<head>` `</head>` рисунок 3.7.

```

<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=UA-141767133-1"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());

  gtag('config', 'UA-141767133-1');
</script>
</head>

```

Рисунок 3.7 - JS-код Google Analytics

### 3.4 CMS платформи та їх різновид

CMS - це система управління контентом, набір скриптів для створення, редагування і управління контентом сайту. На професійному жаргоні CMS також називають «двигун». Прикладами CMS є WordPress, Joomla, PrestaShop.

Якщо раніше більшість сайтів були статичними; і вимагали внесення правок в їх вміст вручну, то зараз динаміка розвитку проектів вимагає готовності швидко реагувати на зміни і впроваджувати їх з максимальною оперативністю. При цьому не всі користувачі хочуть або можуть собі дозволити звертатися до розробників, особливо якщо сайт вимагає постійної роботи над ним.

У свою чергу, системи управління контентом дозволяють користувачам, які не володіють навичками розробки сайтів і знаннями мов програмування, самостійно працювати над створенням і зміною сайту.

Суть роботи CMS укладена в схемі наступним поділом його сайту і його дизайну. Користувачеві надається можливість вибрати шаблон - заготовку, якій заздалегідь визначено оформлення сторінки, і залишається тільки заповнити його потрібною інформацією. Більшість систем управління вмістом ґрунтується на використанні візуального редактора (WYSIWYG - від англ. What You See Is What You Get - «що бачиш, то і отримаєш») - програми, що дозволяє за допомогою інтуїтивно зрозумілого інтерфейсу додавати або змінювати інформацію на сайті. Варто відзначити, що сайт не перебуває з сукупності сторінок як такої, а формується динамічно. Доданий контент зберігається в базі даних, наприклад, MySQL, і використовується при генерації сторінки після отримання відповідного запиту з боку клієнта.

### **Переваги і недоліки**

Переваги CMS:

- просто і зручно використовувати;
- доступний широкий функціонал за рахунок доповнень, тим, розширень;
- сайт можна створити за короткий проміжок часу;
- наявність документації.

Недоліки:

- не підходять для нетипових завдань;
- популярні CMS уразливі;
- необхідно стежити за оновленням CMS і сумісністю версій доповнень;
- підвищене споживання ресурсів, особливо при використанні плагінів.

### **3.4.1 Популярні CMS платформи**

Найбільш популярні платформи це WordPress, Joomla, OpenCart, є багато інших, але це найкращі.

WordPress - це система управління вмістом сайту (CMS) з відкритим вихідним кодом, яка розповсюджується під ліцензією GNU GPL версії 2. Написана на PHP, в якості бази даних використовує MySQL. Сфера застосування - від блогів до досить складних новинних ресурсів і навіть інтернет-магазинів. Вбудована система «тем» і «плагінів» разом з вдалою архітектурою дозволяє конструювати практично будь-які проекти. WordPress забезпечує комфортний і нетрудомісткий процес розробки сайту.[20]

Joomla - система управління вмістом (CMS), написана на мовах PHP і JavaScript, що використовує в якості сховища бази даних СУБД MySQL або інші стандартні промислові реляційні СУБД. Є вільним програмним забезпеченням, поширюваним під ліцензією GNU GPL.[21]

OpenCart - e-commerce platform, орієнтована на створення інтернет-магазинів. Є вільним програмним забезпеченням, поширюваним за ліцензією.

Програмне забезпечення написано на мові програмування PHP, а в архітектурі використаний шаблон проектування MVCL.[22]

Для створення сайту кафедри обрали WordPress. OpenCart для сайту кафедри не підходить, але для створення інтернет магазинів він буде кращім ніж інші. Joomla та WordPress дуже схожі, для створення сайтів, функціонал майже

один в один, але в Joomla рідше бувають оновлення ніж у конкурента WordPress. Також для WordPress величезна кількість плагінів та шаблонів.

### **3.5. Створення сайту за допомогою WordPress**

Щоб збудувати сайт кафедри потрібно створити локальний сервер. Для цього обрали Open Server.

Open Server Panel - це портативна серверна платформа і програмне середовище, створена спеціально для веб-розробників з урахуванням їх рекомендацій і побажань.[23]

Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, має потужні можливості з адміністрування та налаштування компонентів. Платформа широко використовується з метою розробки, налагодження і тестування веб-проектів, а так само для надання веб-сервісів в локальних мережах.

Після встановлення серверу, потрібно завантажити WordPress с сайту на встановити его на сервер. Як завантажили файли на сервер відкриваємо сайт та налаштовуємо (рисунок 3.8).

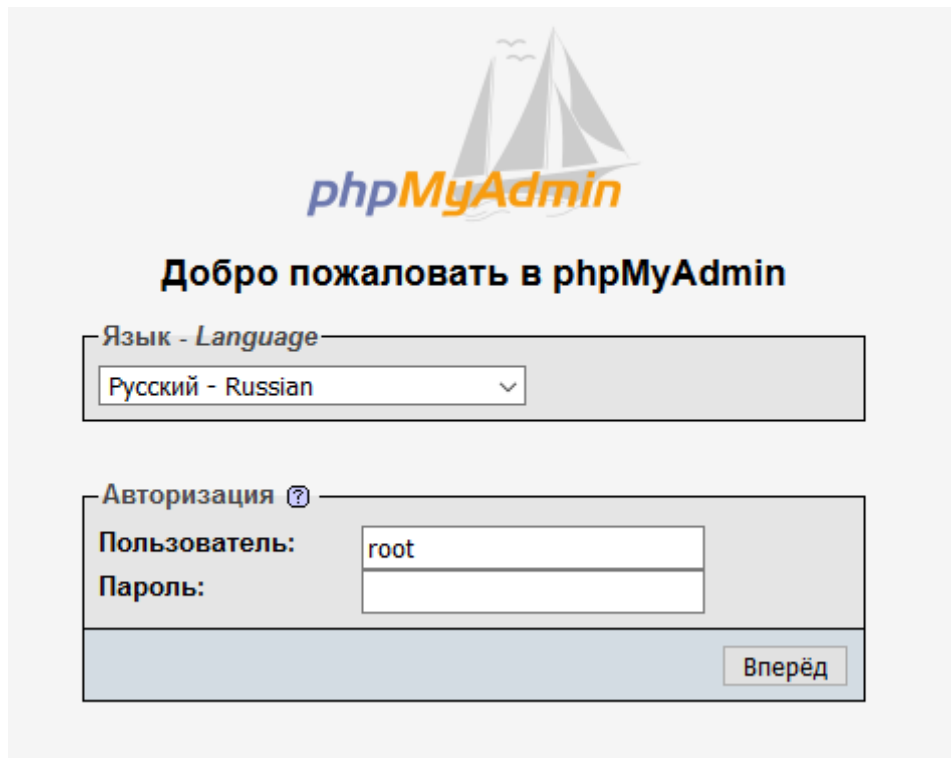


Введите здесь информацию о подключении к базе данных. Если вы в ней не уверены, свяжитесь с хостинг-провайдером.

<b>Имя базы данных</b>	<input type="text" value="wordpress"/>	Имя базы данных, в которую вы хотите установить WordPress.
<b>Имя пользователя</b>	<input type="text" value="username"/>	Имя пользователя базы данных.
<b>Пароль</b>	<input type="text" value="password"/>	Пароль пользователя базы данных.
<b>Сервер базы данных</b>	<input type="text" value="localhost"/>	Если localhost не работает, нужно узнать правильный адрес в службе поддержки хостинг-провайдера.
<b>Префикс таблиц</b>	<input type="text" value="wp_"/>	Если вы хотите запустить несколько копий WordPress в одной базе, измените это значение.

Рисунок 4.8 – наладки сайта

Дали потрібно створити базу даних, для цього зайдіть в панель «php my admin» (Рисунок 3.9).



**phpMyAdmin**

**Добро пожаловать в phpMyAdmin**

Язык - Language

Русский - Russian

Авторизация ?

Пользователь: root

Пароль:

Вперёд

Рисунок 3.9 - «php my admin»

Коли потрапили в панель створюємо новий обліковий запис, заповнюємо базу даних( Рисунок 3.9.1).

**Добавить учетную запись пользователя**

**Информация учётной записи**

Имя пользователя:

Имя хоста:

Пароль:   Strength:

Подтверждение:

Модуль аутентификации:

Создать пароль:

**База данных для учетных записей пользователей**

☒ Создать базу данных с таким же именем и предоставить на неё все привилегии.  
☐ Предоставить все привилегии на то, что подпадает под шаблон (имя пользователя\\_%).

**Глобальные привилегии** ☒ **Отметить все**

Примечание: типы привилегий MySQL отображаются по-английски.

Данные	Структура	Администрирование	Ограничение на использование ресурсов
<input checked="" type="checkbox"/> SELECT <input checked="" type="checkbox"/> INSERT <input checked="" type="checkbox"/> UPDATE <input checked="" type="checkbox"/> DELETE <input checked="" type="checkbox"/> FILE	<input checked="" type="checkbox"/> CREATE <input checked="" type="checkbox"/> ALTER <input checked="" type="checkbox"/> INDEX <input checked="" type="checkbox"/> DROP <input checked="" type="checkbox"/> CREATE TEMPORARY TABLES <input checked="" type="checkbox"/> SHOW VIEW <input checked="" type="checkbox"/> CREATE ROUTINE <input checked="" type="checkbox"/> ALTER ROUTINE <input checked="" type="checkbox"/> EXECUTE <input checked="" type="checkbox"/> CREATE VIEW <input checked="" type="checkbox"/> EVENT <input checked="" type="checkbox"/> TRIGGER	<input checked="" type="checkbox"/> GRANT <input checked="" type="checkbox"/> SUPER <input checked="" type="checkbox"/> PROCESS <input checked="" type="checkbox"/> RELOAD <input checked="" type="checkbox"/> SHUTDOWN <input checked="" type="checkbox"/> SHOW DATABASES <input checked="" type="checkbox"/> LOCK TABLES <input checked="" type="checkbox"/> REFERENCES <input checked="" type="checkbox"/> REPLICATION CLIENT <input checked="" type="checkbox"/> REPLICATION SLAVE <input checked="" type="checkbox"/> CREATE USER	<p><small>Замечание: Установка значения параметров в 0 (ноль), снимает ограничения.</small></p> <p>MAX QUERIES PER HOUR <input type="text" value="0"/></p> <p>MAX UPDATES PER HOUR <input type="text" value="0"/></p> <p>MAX CONNECTIONS PER HOUR <input type="text" value="0"/></p> <p>MAX USER_CONNECTIONS <input type="text" value="0"/></p>

Рисунок 3.9.1 - новый обліковий запис

Як налаштували сайт, потрапляємо на головні налаштування сайту (рисунок 3.9.2).

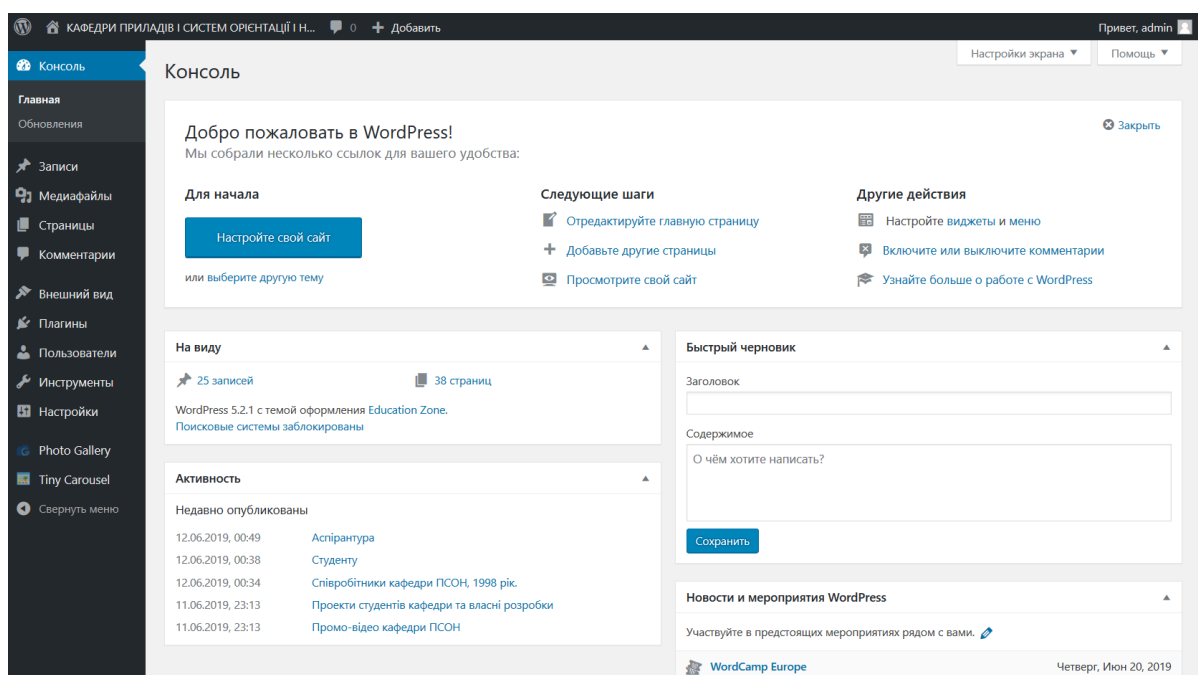


Рисунок 3.9.2 – Головні налаштування сайту

Щоб створити гарний сайт потрібно скористуватися «темой», її можливо встановити з стандартних, чи скачати. Для сайту завантажили тему яка називається «Education Zone» та встановили на додаток. Після того як встановили потрібно налаштувати тему. Налаштування теми виглядають (Рисунок 3.9.3).

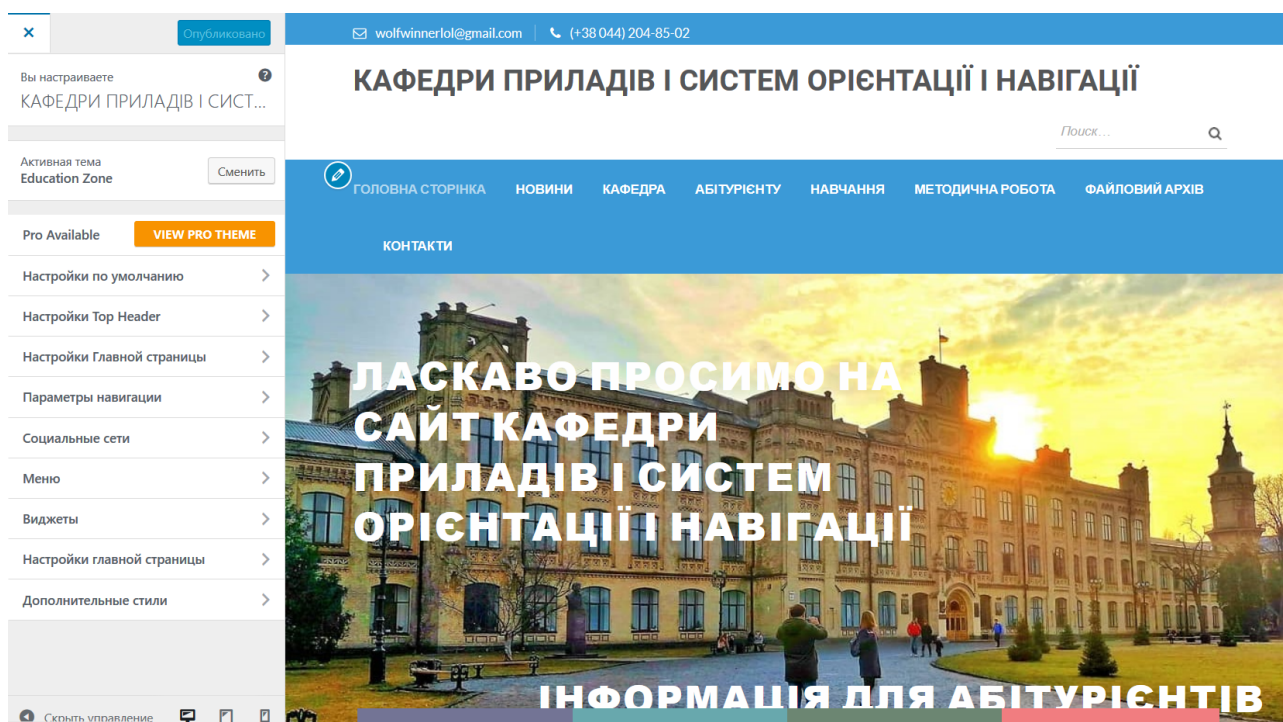


Рисунок 3.9.3 – налаштування теми

На рисунку вище ми бачимо налаштування візуальні, але не все можливо налаштувати візуальним редактором. WordPress передбачає налаштування з втручання в код теми, тому можливо допрацювати тему яку ми завантажили (рисунок 3.9.4).

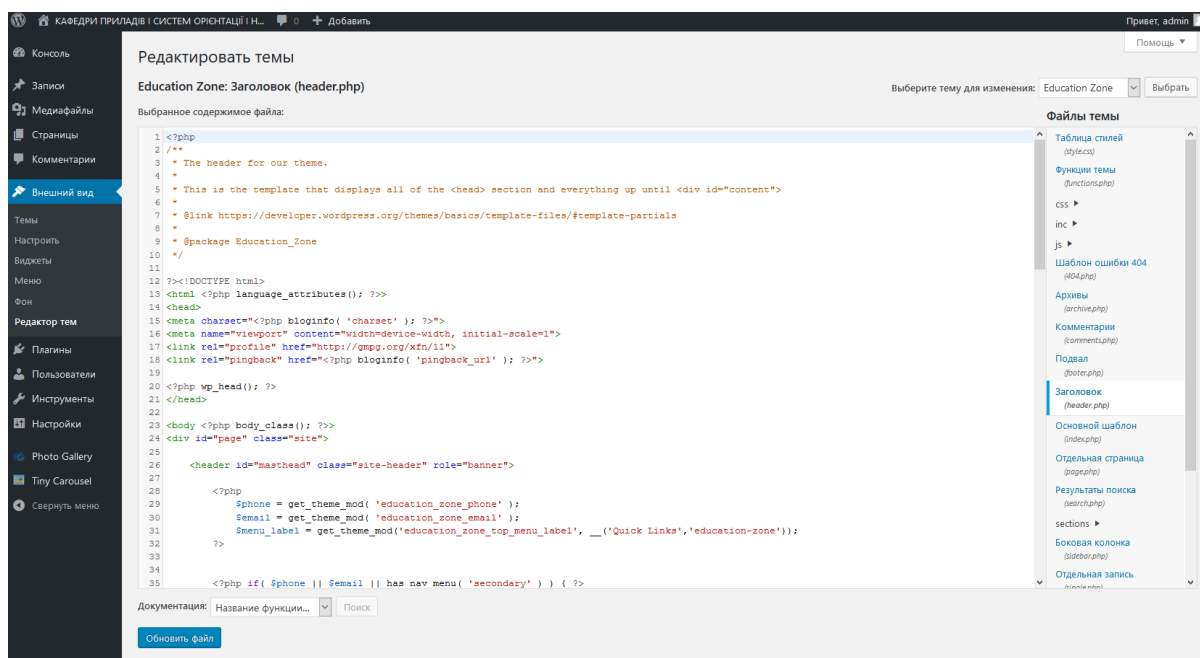


Рисунок 3.9.4 – налаштування теми через код

Після всіх налаштувань, наповнюємо сайт контентом. Тепер на сайт легко кожен має можливість додавати, редагувати чи удаляти інформацію. Тепер сайт виглядає більш привабливим та функціональним (рисунк 3.9.5).

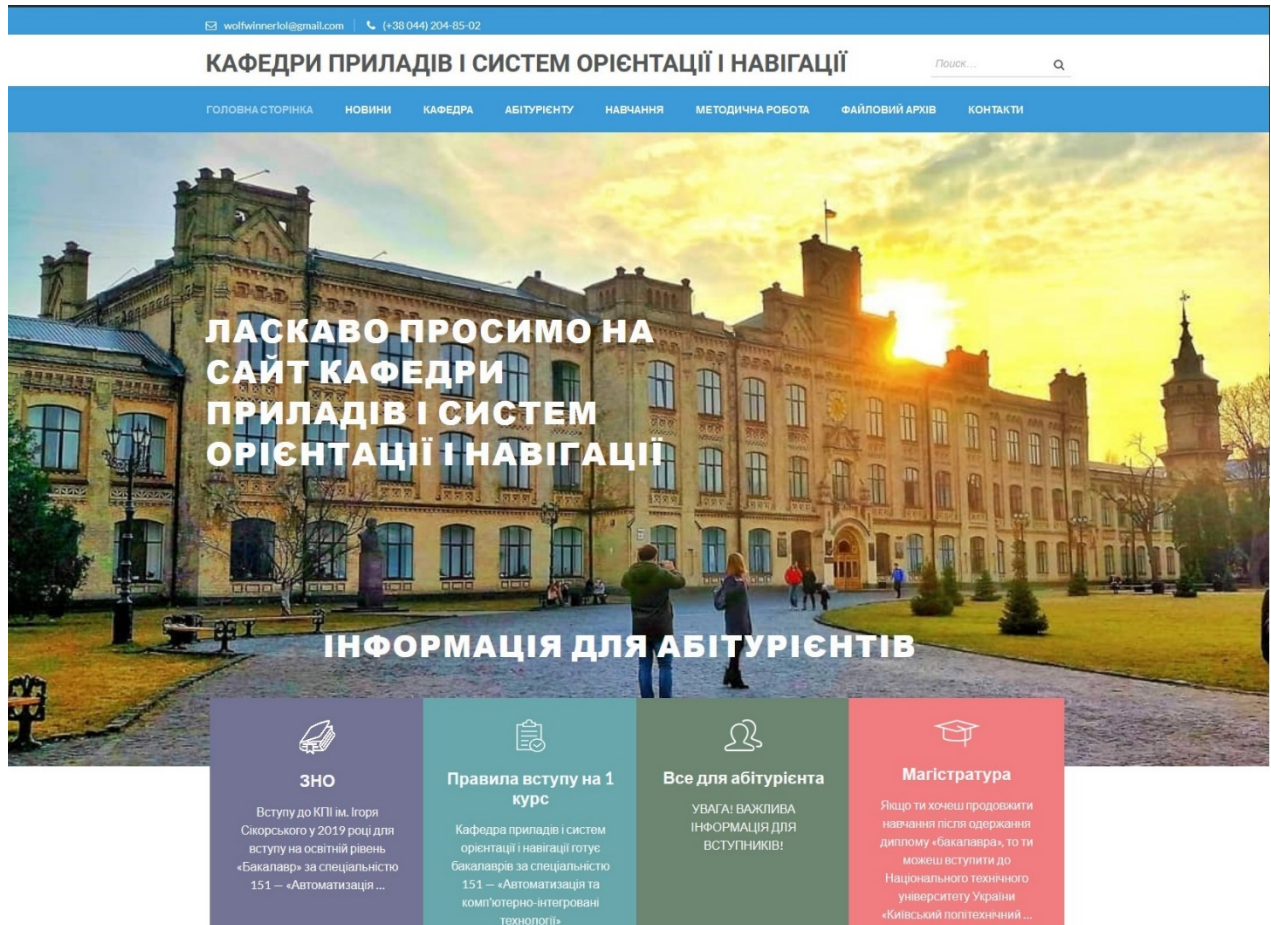


Рисунок 3.9.5– новий сайт кафедри ПСОН

SEO оптимізація, на створених сайтах за допомогою WordPress, проводять завдяки плагінам, яких дуже багато. Плагіни, легко та швидко, ставити на сайт. Вони автоматично провисують мета – теги.

## ВИСНОВКИ

Раніше сайт кафедри був створений тільки за допомогою HTML, CSS, верстка сайту була таблична, а сайт був не адаптивний. Після удосконалення, сайт був створений за допомогою фреймворка Bootstrap, та допоміжними бібліотеками такі як, JQuery, Normalize, JavaScript. Але ці сайти були статичними, управління сайту, було дуже незручне, через те, що потрібно було заходити в код сайту, щоб додавати, видалити чи редагувати. Тому було прийняте рішення, створення сайту на платформі CMS. Для цього обрали WordPress. Тепер сайт став динамічний, та легкий в керуванні контентом, але тема обмежує редагувати дизайну. Також якщо порівнювати SEO оптимізацію на статичному та динамічному сайті, то оптимізація буде краща на статичному, через те, що пошукові системи орієнтуються на завантаження сторінки с сервера до користувача, то статичний швидше буде завантажуватися. Але якщо порівнювати швидкість завантаження, то це майже непомітно.

Підіб'ємо підсумки, для відвідувача сайту немає різниці, на якій платформі, чи який фрейворк використовували, головне, щоб все було зрозуміло. Для адміністрування, вигідно буде, якщо для створення сайту використовують CMS. Тому сайт кафедри був створений на CMS платформі.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Мова гіпертекстової розмітки HTML5. [Електронний ресурс]: <https://html.spec.whatwg.org> – Режим доступу: <https://html.spec.whatwg.org>
2. Історія створення першого сайту в світі. [Електронний ресурс]: <https://www.w3.org/People/Berners-Lee/>
3. Біографія Тім Бернерс-Лі. [Електронний ресурс]: <https://www.w3.org/People/Berners-Lee/>
4. Веб-фреймворки і з чим їх їдять [Електронний ресурс] - Режим доступу до ресурсу: <http://iwsn.ru/blog/show/veb-freymvorki-i-s-chem-ih-edyat>.
5. Фреймворки в веб-розробці [Електронний ресурс] - Режим доступу до ресурсу: [https://web-creator.ru/articles/about\\_frameworks](https://web-creator.ru/articles/about_frameworks).
6. Результати тестування шести провідних фреймворків на продуктивність [Електронний ресурс] - Режим доступу до ресурсу: <http://www.alrond.com/ru/2007/jan/25/rezultaty-testirovaniya-6-frameworks/>.
7. Адаптивні CSS-фреймворки, сітки, класи видимості [Електронний ресурс] - Режим доступу до ресурсу: <http://klondike-studio.ru/blog/responsive-cssframework/>.
8. Огляд CSS-фреймворків [Електронний ресурс] - Режим доступу до ресурсу: <http://iantonov.me/page/obzor-css-freymvorkov>.
9. Використання PHP фреймворків в розробці сайту [Електронний ресурс] - Режим доступу до ресурсу: <http://ukrbukva.net/page,5,39718-Ispol-zovanie-PHPfreymvorkov-v-razrabotke-saiyta.html>.
10. Порівняння каркасів веб-додатків [Електронний ресурс] - Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/>.
11. Огляди Web-фреймворків [Електронний ресурс] - Режим доступу до ресурсу: <https://praktikatech.wordpress.com/category/обзоры-web-фреймворков/>

12. Огляд CSS-фреймворків [Електронний ресурс] - Режим доступу до ресурсу: <http://iantonov.me/page/obzor-css-frejmworkov>.
13. Повне керівництво по Yii [Електронний ресурс] - Режим доступу до ресурсу: <http://www.yiiframework.com/doc/guide/1.1/ru/index>.
14. Apache HTTP-сервер [Електронний ресурс]: <http://httpd.apache.org/>
15. SEO або пошукова оптимізація сайту [Електронний ресурс]: <https://support.google.com/webmasters/answer/35769?hlrm=en>
16. Мета-теги поміщаються всередину HTML-документа [Електронний ресурс]: <http://vancouver-webpages.com/META/metatags.detail.html>
17. Tag manager [Електронний ресурс]: <https://www.owox.ru/blog/use-cases/what-is-google-tag-manager/>
18. Google Optimize [Електронний ресурс]: <https://pr-cy.ru/news/p/6336>
19. WordPress [Електронний ресурс]: <https://wordpress.co.ua/stvoryty-blog-na-wordpress/3-blog-management-system-wordpress>
20. WordPress [Електронний ресурс]: <https://wordpress.co.ua/stvoryty-blog-na-wordpress/3-blog-management-system-wordpress>
21. Joomla [Електронний ресурс]: <http://joomla.ru/docs/administrator/joomla3-start/1742-chto-takoe-joomla>
22. OpenCart [Електронний ресурс]: <http://docs.opencart.com/en-gb/requirements/>
23. Open Server Panel [Електронний ресурс]: <https://ospanel.io/docs/>